

***Bugs* czyli pluskwy**



***Bugs* czyli pluskwy**



**Nie ma programu bez błędów.
Są tylko takie, w których błędu jeszcze nie wykryto.
(ludowa mądrość programistów)**

***Bugs* czyli pluskwy**



Nie ma programu bez błędów.

Są tylko takie, w których błędu jeszcze nie wykryto.

(ludowa mądrość programistów)

Błąd w programie jest chlebem codziennym programisty i nie jest rzeczą wstydliwą.

***Bugs* czyli pluskwy**



Nie ma programu bez błędów.

Są tylko takie, w których błędu jeszcze nie wykryto.

(ludowa mądrość programistów)

Błąd w programie jest chlebem codziennym programisty i nie jest rzeczą wstydliwą. Wstydzić należy się tylko **bezradności** w obliczu błędów.

Trzeba umieć radzić sobie z błędami.

***Bugs* czyli pluskwy**



Nie ma programu bez błędów.

Są tylko takie, w których błędy jeszcze nie wykryto.

(ludowa mądrość programistów)

Błąd w programie jest chlebem codziennym programisty i nie jest rzeczą wstydliwą. Wstydzić należy się tylko **bezradności** w obliczu błędów.

Trzeba umieć radzić sobie z błędami.

**Testowanie może pomóc w znalezieniu błędu,
ale nie może dowodzić braku błędów.**

(inna ludowa mądrość programistów)

Bugs czyli pluskwy



**Nie ma programu bez błędów.
Są tylko takie, w których błędy jeszcze nie wykryto.
(ludowa mądrość programistów)**

Błąd w programie jest chlebem codziennym programisty i nie jest rzeczą wstydliwą. Wstydzić należy się tylko **bezradności** w obliczu błędów.

Trzeba umieć radzić sobie z błędami.

**Testowanie może pomóc w znalezieniu błędu,
ale nie może dowodzić braku błędów.
(inna ludowa mądrość programistów)**

Każdy „gotowy” program należy poddać serii krytycznych testów (*crash-test* — test na zderzenie).

Bugs czyli pluskwy



Nie ma programu bez błędów.

Są tylko takie, w których błędy jeszcze nie wykryto.

(ludowa mądrość programistów)

Błąd w programie jest chlebem codziennym programisty i nie jest rzeczą wstydliwą. Wstydzić należy się tylko **bezradności** w obliczu błędów.

Trzeba umieć radzić sobie z błędami.

**Testowanie może pomóc w znalezieniu błędu,
ale nie może dowodzić braku błędów.**

(inna ludowa mądrość programistów)

Każdy „gotowy” program należy poddać serii krytycznych testów (*crash-test* — test na zderzenie).

Najprostsze błędy są sygnalizowane w trakcie kompilacji ze wskazaniem numeru wiersza

Bugs czyli pluskwy



Nie ma programu bez błędów.

Są tylko takie, w których błędu jeszcze nie wykryto.

(ludowa mądrość programistów)

Błąd w programie jest chlebem codziennym programisty i nie jest rzeczą wstydliwą. Wstydzić należy się tylko **bezradności** w obliczu błędów.

Trzeba umieć radzić sobie z błędami.

**Testowanie może pomóc w znalezieniu błędu,
ale nie może dowodzić braku błędów.**

(inna ludowa mądrość programistów)

Każdy „gotowy” program należy poddać serii krytycznych testów (*crash-test* — test na zderzenie).

Najprostsze błędy są sygnalizowane w trakcie kompilacji ze wskazaniem numeru wiersza — ale brak takich sygnałów **nie stanowi** dowodu bezbłędności programu.

***Bugs* czyli pluskwy**

Przykład: (liczenie średnich)

***Bugs* czyli pluskwy**

Przykład: (liczenie średnich)

Średnia arytmetyczna:
$$\frac{a_0 + a_1 + \dots + a_{n-1}}{n}$$

***Bugs* czyli pluskwy**

Przykład: (liczenie średnich)

Średnia arytmetyczna:
$$\frac{a_0 + a_1 + \dots + a_{n-1}}{n}$$

Średnia geometryczna:
$$\sqrt[n]{a_0 \cdot a_1 \cdot \dots \cdot a_{n-1}}$$

Bugs czyli pluskwy

Przykład: (liczenie średnich)

Średnia arytmetyczna:
$$\frac{a_0 + a_1 + \dots + a_{n-1}}{n}$$

Średnia geometryczna:
$$\sqrt[n]{a_0 \cdot a_1 \cdot \dots \cdot a_{n-1}}$$

Średnia harmoniczna:
$$\frac{n}{\frac{1}{a_0} + \frac{1}{a_1} + \dots + \frac{1}{a_{n-1}}}$$

Bugs czyli pluskwy

Przykład: (liczenie średnich)

Średnia arytmetyczna:
$$\frac{a_0 + a_1 + \dots + a_{n-1}}{n}$$

Średnia geometryczna:
$$\sqrt[n]{a_0 \cdot a_1 \cdot \dots \cdot a_{n-1}}$$

Średnia harmoniczna:
$$\frac{n}{\frac{1}{a_0} + \frac{1}{a_1} + \dots + \frac{1}{a_{n-1}}}$$

Program ma wczytywać po jednej liczbie rzeczywistej

Bugs czyli pluskwy

Przykład: (liczenie średnich)

Średnia arytmetyczna:
$$\frac{a_0 + a_1 + \dots + a_{n-1}}{n}$$

Średnia geometryczna:
$$\sqrt[n]{a_0 \cdot a_1 \cdot \dots \cdot a_{n-1}}$$

Średnia harmoniczna:
$$\frac{n}{\frac{1}{a_0} + \frac{1}{a_1} + \dots + \frac{1}{a_{n-1}}}$$

Program ma wczytywać po jednej liczbie rzeczywistej i drukować trzy średnie dotąd wczytanych liczb

Bugs czyli pluskwy

Przykład: (liczenie średnich)

Średnia arytmetyczna:
$$\frac{a_0 + a_1 + \dots + a_{n-1}}{n}$$

Średnia geometryczna:
$$\sqrt[n]{a_0 \cdot a_1 \cdot \dots \cdot a_{n-1}}$$

Średnia harmoniczna:
$$\frac{n}{\frac{1}{a_0} + \frac{1}{a_1} + \dots + \frac{1}{a_{n-1}}}$$

Program ma wczytywać po jednej liczbie rzeczywistej i drukować trzy średnie dotąd wczytanych liczb; następnie pytać, czy będzie ich więcej.

***Bugs* czyli pluskwy**

Średnia arytmetyczna:
$$\frac{a_0 + a_1 + \dots + a_{n-1}}{n}$$

Średnia geometryczna:
$$\sqrt[n]{a_0 \cdot a_1 \cdot \dots \cdot a_{n-1}}$$

Średnia harmoniczna:
$$\frac{n}{\frac{1}{a_0} + \frac{1}{a_1} + \dots + \frac{1}{a_{n-1}}}$$

Bugs czyli pluskwy

Przykład: (liczenie średnich)

Średnia arytmetyczna: $\frac{a_0 + a_1 + \dots + a_{n-1}}{n}$

Średnia geometryczna: $\sqrt[n]{a_0 \cdot a_1 \cdot \dots \cdot a_{n-1}}$

Średnia harmoniczna: $\frac{n}{\frac{1}{a_0} + \frac{1}{a_1} + \dots + \frac{1}{a_{n-1}}}$

```
n=0; suma=0; iloczyn=1; suma_odwr=0;
```

Bugs czyli pluskwy

Przykład: (liczenie średnich)

Średnia arytmetyczna: $\frac{a_0 + a_1 + \dots + a_{n-1}}{n}$

Średnia geometryczna: $\sqrt[n]{a_0 \cdot a_1 \cdot \dots \cdot a_{n-1}}$

Średnia harmoniczna: $\frac{n}{\frac{1}{a_0} + \frac{1}{a_1} + \dots + \frac{1}{a_{n-1}}}$

```
n=0; suma=0; iloczyn=1; suma_odwr=0;
zapytaj, czy będzie więcej danych;
while (będzie ich więcej) {

    zapytaj, czy będzie więcej danych;
}
```

Bugs czyli pluskwy

Przykład: (liczenie średnich)

Średnia arytmetyczna: $\frac{a_0 + a_1 + \dots + a_{n-1}}{n}$

Średnia geometryczna: $\sqrt[n]{a_0 \cdot a_1 \cdot \dots \cdot a_{n-1}}$

Średnia harmoniczna: $\frac{n}{\frac{1}{a_0} + \frac{1}{a_1} + \dots + \frac{1}{a_{n-1}}}$

```
n=0; suma=0; iloczyn=1; suma_odwr=0;
zapytaj, czy będzie więcej danych;
while (będzie ich więcej) {
    wczytaj liczbę do nowa; n++;

    zapytaj, czy będzie więcej danych;
}
```

Bugs czyli pluskwy

Przykład: (liczenie średnich)

Średnia arytmetyczna: $\frac{a_0 + a_1 + \dots + a_{n-1}}{n}$

Średnia geometryczna: $\sqrt[n]{a_0 \cdot a_1 \cdot \dots \cdot a_{n-1}}$

Średnia harmoniczna: $\frac{n}{\frac{1}{a_0} + \frac{1}{a_1} + \dots + \frac{1}{a_{n-1}}}$

```
n=0; suma=0; iloczyn=1; suma_odwr=0;
zapytaj, czy będzie więcej danych;
while (będzie ich więcej) {
    wczytaj liczbę do nowa; n++;
    suma += nowa;

    zapytaj, czy będzie więcej danych;
}
```

Bugs czyli pluskwy

Przykład: (liczenie średnich)

Średnia arytmetyczna:
$$\frac{a_0 + a_1 + \dots + a_{n-1}}{n}$$

Średnia geometryczna:
$$\sqrt[n]{a_0 \cdot a_1 \cdot \dots \cdot a_{n-1}}$$

Średnia harmoniczna:
$$\frac{n}{\frac{1}{a_0} + \frac{1}{a_1} + \dots + \frac{1}{a_{n-1}}}$$

```
n=0; suma=0; iloczyn=1; suma_odwr=0;
zapytaj, czy będzie więcej danych;
while (będzie ich więcej) {
    wczytaj liczbę do nowa; n++;
    suma += nowa; wydrukuj suma/n;

    zapytaj, czy będzie więcej danych;
}
```

Bugs czyli pluskwy

Przykład: (liczenie średnich)

Średnia arytmetyczna: $\frac{a_0 + a_1 + \dots + a_{n-1}}{n}$

Średnia geometryczna: $\sqrt[n]{a_0 \cdot a_1 \cdot \dots \cdot a_{n-1}}$

Średnia harmoniczna: $\frac{n}{\frac{1}{a_0} + \frac{1}{a_1} + \dots + \frac{1}{a_{n-1}}}$

```
n=0; suma=0; iloczyn=1; suma_odwr=0;
zapytaj, czy będzie więcej danych;
while (będzie ich więcej) {
    wczytaj liczbę do nowa; n++;
    suma += nowa; wydrukuj suma/n;
    iloczyn *= nowa; wydrukuj iloczyn1/n;

    zapytaj, czy będzie więcej danych;
}
```

Bugs czyli pluskwy

Przykład: (liczenie średnich)

Średnia arytmetyczna:
$$\frac{a_0 + a_1 + \dots + a_{n-1}}{n}$$

Średnia geometryczna:
$$\sqrt[n]{a_0 \cdot a_1 \cdot \dots \cdot a_{n-1}}$$

Średnia harmoniczna:
$$\frac{n}{\frac{1}{a_0} + \frac{1}{a_1} + \dots + \frac{1}{a_{n-1}}}$$

```
n=0; suma=0; iloczyn=1; suma_odwr=0;
zapytaj, czy będzie więcej danych;
while (będzie ich więcej) {
    wczytaj liczbę do nowa; n++;
    suma += nowa; wydrukuj suma/n;
    iloczyn *= nowa; wydrukuj iloczyn1/n;
    suma_odwr += 1/nowa; wydrukuj n/suma_odwr;
    zapytaj, czy będzie więcej danych;
}
```

Bugs czyli pluskwy

Przykład: (liczenie średnich)

```
n=0; suma=0; iloczyn=1; suma_odwr=0;

zapytaj, czy będzie więcej danych;

while (będzie ich więcej) {

    wczytaj liczbę do nowa; n++;

    suma += nowa; wydrukuj suma/n;

    iloczyn *= nowa; wydrukuj iloczyn1/n;

    suma_odwr += 1/nowa; wydrukuj n/suma_odwr;

    zapytaj, czy będzie więcej danych;

}
```

Bugs czyli pluskwy

Przykład: (liczenie średnich)

```
printf("Wiecej danych? (Tak/Nie) ");  
do {odp = getchar();} while (isspace(odp));
```

```
n=0; suma=0; iloczyn=1; suma_odwr=0;
```

```
zapytaj, czy będzie więcej danych;
```

```
while (będzie ich więcej) {
```

```
    wczytaj liczbę do nowa; n++;
```

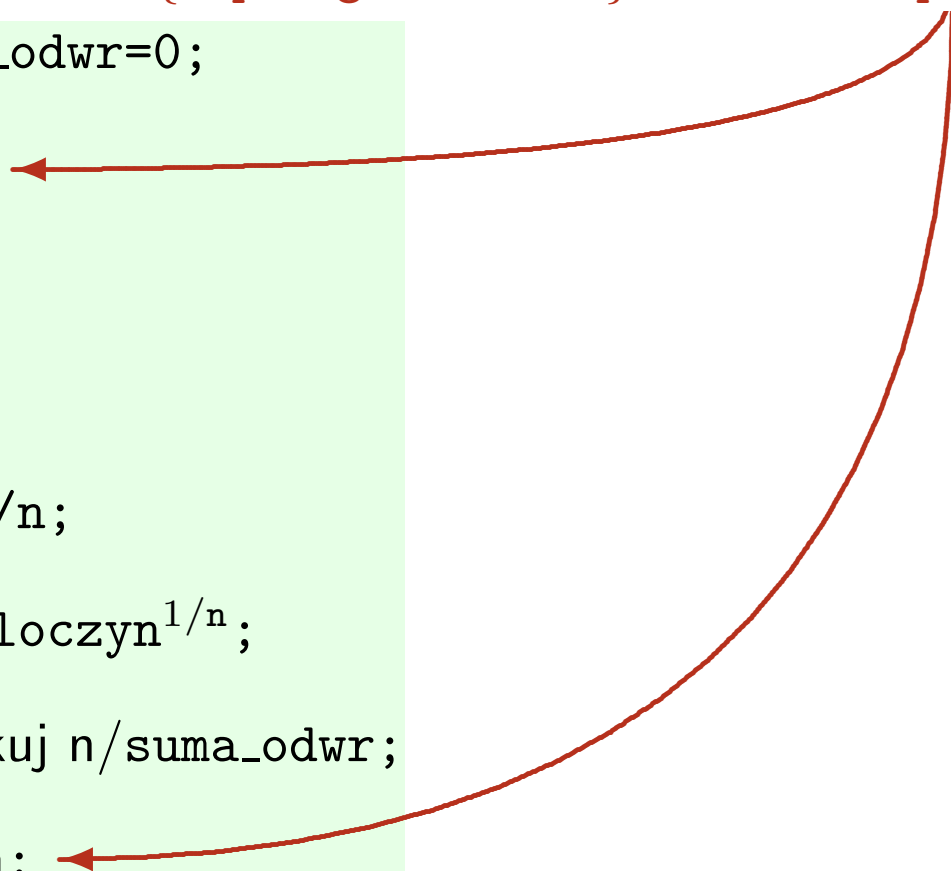
```
    suma += nowa; wydrukuj suma/n;
```

```
    iloczyn *= nowa; wydrukuj iloczyn1/n;
```

```
    suma_odwr += 1/nowa; wydrukuj n/suma_odwr;
```

```
    zapytaj, czy będzie więcej danych;
```

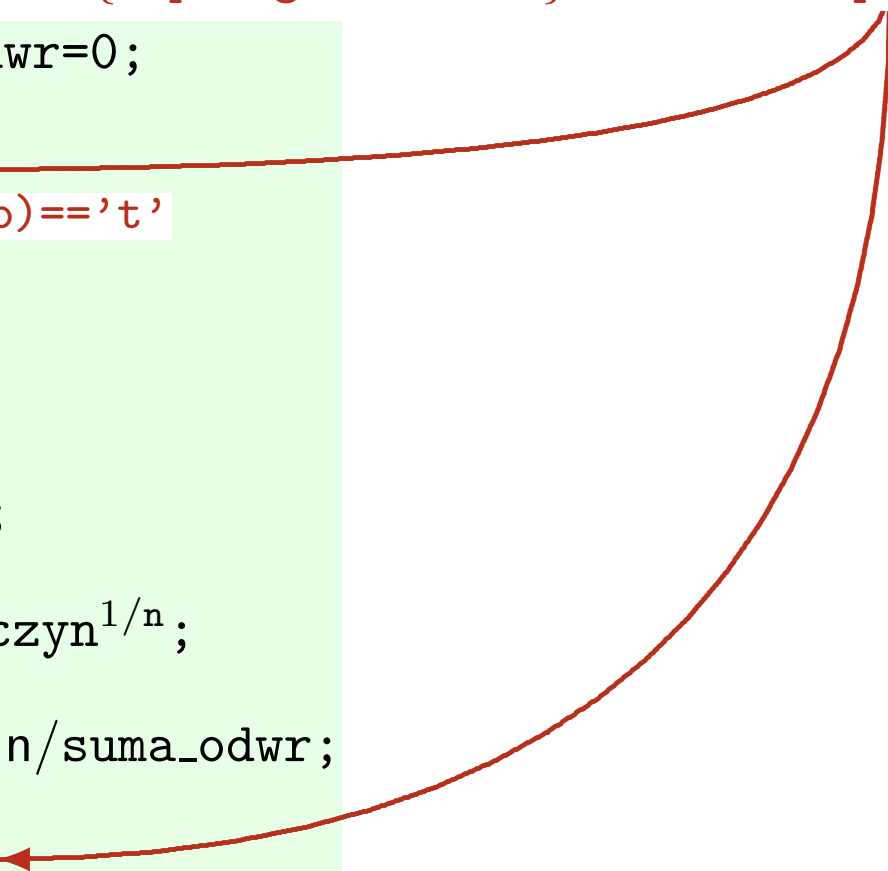
```
}
```



Bugs czyli pluskwy

Przykład: (liczenie średnich)

```
printf("Wiecej danych? (Tak/Nie) ");  
do {odp = getchar();} while (isspace(odp));  
  
n=0; suma=0; iloczyn=1; suma_odwr=0;  
  
zapytaj, czy będzie więcej danych; ←  
    tolower(odp)=='t'  
while (będzie ich więcej) {  
    wczytaj liczbę do nowa; n++;  
  
    suma += nowa; wydrukuj suma/n;  
  
    iloczyn *= nowa; wydrukuj iloczyn1/n;  
  
    suma_odwr += 1/nowa; wydrukuj n/suma_odwr;  
  
    zapytaj, czy będzie więcej danych; ←  
}
```



Bugs czyli pluskwy

Przykład: (liczenie średnich)

```

printf("Wiecej danych? (Tak/Nie) ");
do {odp = getchar();} while (isspace(odp));

n=0; suma=0; iloczyn=1; suma_odwr=0;

zapytaj, czy będzie więcej danych;
while (będzie ich więcej) {
    wczytaj liczbę do nowa; n++;
    suma += nowa; wydrukuj suma/n;
    iloczyn *= nowa; wydrukuj iloczyn1/n;
    suma_odwr += 1/nowa; wydrukuj n/suma_odwr;
    zapytaj, czy będzie więcej danych;
}

```

The diagram illustrates a loop structure. A red arrow originates from the end of the loop body (after the 'zapytaj, czy będzie więcej danych;' statement) and points back to the start of the loop body (before the 'wczytaj liczbę do nowa;' statement). This represents a loop iteration.

Bugs czyli pluskwy

Przykład: (liczenie średnich)

```

printf("Wiecej danych? (Tak/Nie) ");
do {odp = getchar();} while (isspace(odp));

n=0; suma=0; iloczyn=1; suma_odwr=0;

zapytaj, czy będzie więcej danych;
while (będzie ich więcej) {
    wczytaj liczbę do nowa; n++;
    suma += nowa; wydrukuj suma/n;
    iloczyn *= nowa; wydrukuj iloczyn1/n;
    suma_odwr += 1/nowa; wydrukuj n/suma_odwr;
    zapytaj, czy będzie więcej danych;
}

```

Diagram illustrating the flow of execution between the two 'zapytaj, czy będzie więcej danych;' statements. A red arrow points from the first statement to the second, and another red arrow points from the second statement back to the first, forming a loop.

Annotations in the code:

- `tolower(odp)=='t'` is annotated with a red arrow pointing to the condition of the while loop.
- `pow(iloczyn,1/n);` is annotated with a red arrow pointing to the expression `iloczyn1/n`.

***Bugs* czyli pluskwy**

Przykład: (liczenie średnich — **BŁĘDY** wykrywane przy kompilacji)

Bugs czyli pluskwy

Przykład: (liczenie średnich — **BŁĘDY** wykrywane przy kompilacji)

kompilacja programu:
`gcc -Wall nazwa_programu.c`

```
srednie.c:5:3: error: unknown type name 'duble'; did you mean 'double'?  
  duble suma=0.0, iloczyn=1..0, suma_odwr=0.0, nowa;  
  ^~~~~  
  double
```

Bugs czyli pluskwy

Przykład: (liczenie średnich — **BŁĘDY** wykrywane przy kompilacji)

kompilacja programu:
gcc -Wall nazwa_programu.c

plik

srednie.c:5:3: error: unknown type name 'duble'; did you mean 'double'?
 duble suma=0.0, iloczyn=1..0, suma_odwr=0.0, nowa;
 ^~~~~
 double

Bugs czyli pluskwy

Przykład: (liczenie średnich — **BŁĘDY** wykrywane przy kompilacji)

kompilacja programu:
`gcc -Wall nazwa_programu.c`

numer wiersza

numer znaku w wierszu

plik

```
srednie.c:5:3: error: unknown type name 'duble'; did you mean 'double'?  
  duble suma=0.0, iloczyn=1..0, suma_odwr=0.0, nowa;  
  ^~~~~  
  double
```

Bugs czyli pluskwy

Przykład: (liczenie średnich — **BŁĘDY** wykrywane przy kompilacji)

kompilacja programu:
`gcc -Wall nazwa_programu.c`

numer wiersza
numer znaku w wierszu
plik
error/warning

```
srednie.c:5:3: error: unknown type name 'duble'; did you mean 'double'?  
  duble suma=0.0, iloczyn=1..0, suma_odwr=0.0, nowa;  
  ^~~~~  
  double
```

Bugs czyli pluskwy

Przykład: (liczenie średnich — **BŁĘDY** wykrywane przy kompilacji)

kompilacja programu:
`gcc -Wall nazwa_programu.c`

Diagram illustrating the components of a compiler error message:

- plik** (file): `srednie.c`
- numer wiersza** (line number): `5`
- numer znaku w wierszu** (character position in line): `3`
- error/warning**: `error`
- opis błędu** (error description): `unknown type name 'duble'; did you mean 'double'?`

The error message text is:

```
srednie.c:5:3: error: unknown type name 'duble'; did you mean 'double'?  
  duble suma=0.0, iloczyn=1..0, suma_odwr=0.0, nowa;  
  ^~~~~  
  double
```

Bugs czyli pluskwy

Przykład: (liczenie średnich — **BŁĘDY** wykrywane przy kompilacji)

kompilacja programu:
gcc -Wall nazwa_programu.c

Diagram illustrating the components of a compiler error message:

- plik** (file): srednie.c
- numer wiersza** (line number): 5
- numer znaku w wierszu** (character position in line): 3
- error/warning**: error
- opis błędu** (error description): unknown type name 'duble'; did you mean 'double'?
- sugestia** (suggestion): double

The error message text is:

```
srednie.c:5:3: error: unknown type name 'duble'; did you mean 'double'?
  duble suma=0.0, iloczyn=1..0, suma_odwr=0.0, nowa;
  ^~~~~
  double
```

Bugs czyli pluskwy

Przykład: (liczenie średnich — **BŁĘDY** wykrywane przy kompilacji)

kompilacja programu:
gcc -Wall nazwa_programu.c

numer wiersza numer znaku w wierszu error/warning opis błędu sugestia

plik

```
srednie.c:5:3: error: unknown type name 'duble'; did you mean 'double'?
  duble suma=0.0, iloczyn=1..0, suma_odwr=0.0, nowa;
  ^~~~~
  double
```

zacytowany błędny wiersz

Bugs czyli pluskwy

Przykład: (liczenie średnich — **BŁĘDY** wykrywane przy kompilacji)

kompilacja programu:
gcc -Wall nazwa_programu.c

Diagram illustrating the structure of a compiler error message with annotations:

```
srednie.c:5:3: error: unknown type name 'duble'; did you mean 'double'?
  duble suma=0.0, iloczyn=1..0, suma_odwr=0.0, nowa;
  ^~~~~
  double
```

Annotations and their corresponding parts of the error message:

- plik** (file): srednie.c
- numer wiersza** (line number): 5
- numer znaku w wierszu** (character position): 3
- error/warning**: error
- opis błędu** (error description): unknown type name 'duble'; did you mean 'double'?
- sugestia** (suggestion): did you mean 'double'?
- zacytowany błędny wiersz** (quoted erroneous line): duble suma=0.0, iloczyn=1..0, suma_odwr=0.0, nowa;
- lokalizacja błędu w wierszu** (error location in line): ^~~~~
- double**: double

Bugs czyli pluskwy

Przykład: (liczenie średnich — **BŁĘDY** wykrywane przy kompilacji)

kompilacja programu:
gcc -Wall nazwa_programu.c

numer wiersza
numer znaku w wierszu
plik
error/warning
opis błędu
sugestia
zacytowany błędny wiersz
lokalizacja błędu w wierszu

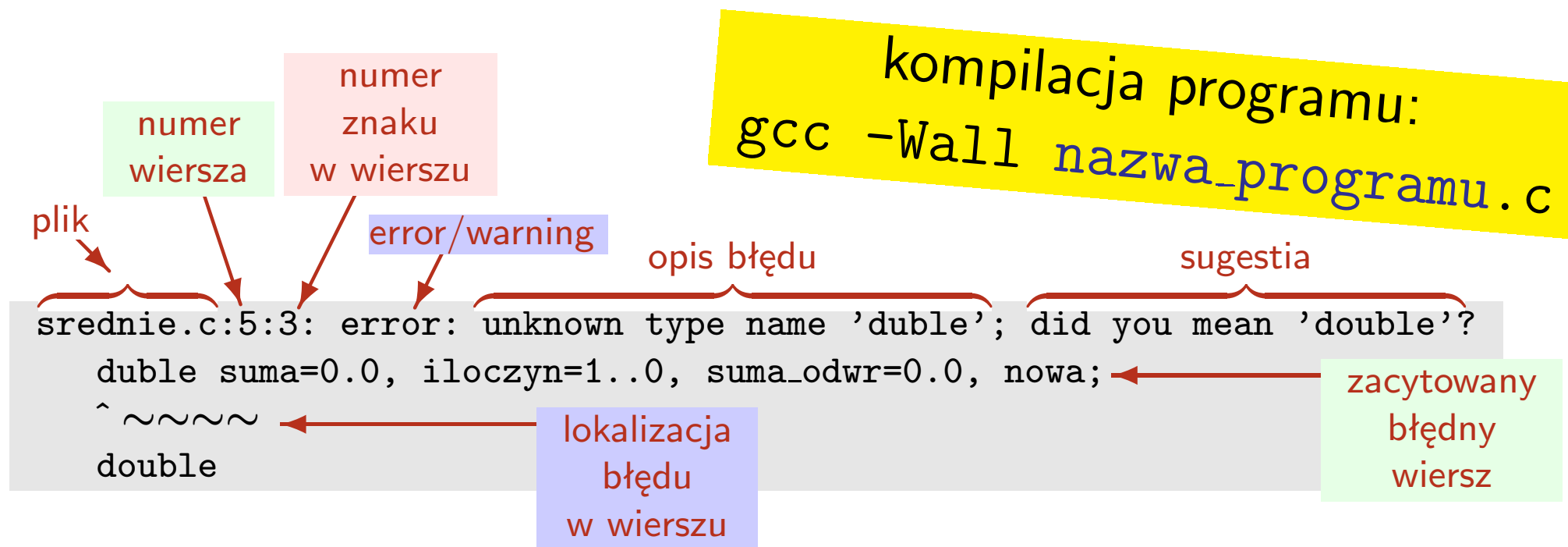
```
srednie.c:5:3: error: unknown type name 'duble'; did you mean 'double'?
  duble suma=0.0, iloczyn=1..0, suma_odwr=0.0, nowa;
  ^~~~~
  double
```

1. W przypadku **błędu** (error) kompilator nie tworzy przekładu

Bugs czyli pluskwy

Przykład: (liczenie średnich — **BŁĘDY** wykrywane przy kompilacji)

kompilacja programu:
gcc -Wall nazwa_programu.c



1. W przypadku **błędu** (error) kompilator nie tworzy przekładu; w przypadku **ostrzeżenia** (warning) tworzy przekład, ale często błędny.

Bugs czyli pluskwy

Przykład: (liczenie średnich — **BŁĘDY** wykrywane przy kompilacji)

kompilacja programu:
gcc -Wall nazwa_programu.c

numer wiersza
numer znaku w wierszu
plik
error/warning
opis błędu
sugestia
zacytowany błędny wiersz
lokalizacja błędu w wierszu

```
srednie.c:5:3: error: unknown type name 'duble'; did you mean 'double'?
  duble suma=0.0, iloczyn=1..0, suma_odwr=0.0, nowa;
  ^~~~~
  double
```

1. W przypadku **błędu** (error) kompilator nie tworzy przekładu; w przypadku **ostrzeżenia** (warning) tworzy przekład, ale często błędny.
2. Numer wiersza dotyczy miejsca, gdzie był kompilator, kiedy wykrył błąd.

Bugs czyli pluskwy

Przykład: (liczenie średnich — **BŁĘDY** wykrywane przy kompilacji)

kompilacja programu:
gcc -Wall nazwa_programu.c

plik numer wiersza numer znaku w wierszu error/warning opis błędu sugestia

```
srednie.c:5:3: error: unknown type name 'duble'; did you mean 'double'?
  duble suma=0.0, iloczyn=1..0, suma_odwr=0.0, nowa;
  ^~~~~
  double
```

zacytowany błędny wiersz

lokalizacja błędu w wierszu

1. W przypadku **błędu** (error) kompilator nie tworzy przekładu; w przypadku **ostrzeżenia** (warning) tworzy przekład, ale często błędny.
2. Numer wiersza dotyczy miejsca, gdzie był kompilator, kiedy wykrył błąd. Sam błąd może znajdować się wcześniej.

Bugs czyli pluskwy

Przykład: (liczenie średnich — **BŁĘDY** wykrywane przy kompilacji)

kompilacja programu:
gcc -Wall nazwa_programu.c

plik numer wiersza numer znaku w wierszu error/warning opis błędu sugestia

```
srednie.c:5:3: error: unknown type name 'duble'; did you mean 'double'?
  duble suma=0.0, iloczyn=1..0, suma_odwr=0.0, nowa;
  ^~~~~
  double
```

zacytowany błędny wiersz

lokalizacja błędu w wierszu

1. W przypadku **błędu** (error) kompilator nie tworzy przekładu; w przypadku **ostrzeżenia** (warning) tworzy przekład, ale często błędny.
2. Numer wiersza dotyczy miejsca, gdzie był kompilator, kiedy wykrył błąd. Sam błąd może znajdować się wcześniej.
3. Po wykryciu błędu kompilator prowizorycznie go „naprawia”, żeby móc iść dalej.

Bugs czyli pluskwy

Przykład: (liczenie średnich — **BŁĘDY** wykrywane przy kompilacji)

kompilacja programu:
gcc -Wall nazwa_programu.c

plik numer wiersza numer znaku w wierszu error/warning opis błędu sugestia

```
srednie.c:5:3: error: unknown type name 'duble'; did you mean 'double'?
  duble suma=0.0, iloczyn=1..0, suma_odwr=0.0, nowa;
  ^~~~~
  double
```

zacytowany błędny wiersz

lokalizacja błędu w wierszu

1. W przypadku **błędu** (error) kompilator nie tworzy przekładu; w przypadku **ostrzeżenia** (warning) tworzy przekład, ale często błędny.
2. Numer wiersza dotyczy miejsca, gdzie był kompilator, kiedy wykrył błąd. Sam błąd może znajdować się wcześniej.
3. Po wykryciu błędu kompilator prowizorycznie go „naprawia”, żeby móc iść dalej. To naprawianie może spowodować lawinę innych błędów.

Bugs czyli pluskwy

Przykład: (liczenie średnich — **BŁĘDY** wykrywane przy kompilacji)

kompilacja programu:
gcc -Wall nazwa_programu.c

plik numer wiersza numer znaku w wierszu error/warning opis błędu sugestia

```
srednie.c:5:3: error: unknown type name 'duble'; did you mean 'double'?
  duble suma=0.0, iloczyn=1..0, suma_odwr=0.0, nowa;
  ^~~~~
  double
```

zacytowany błędny wiersz

lokalizacja błędu w wierszu

1. W przypadku **błędu** (error) kompilator nie tworzy przekładu; w przypadku **ostrzeżenia** (warning) tworzy przekład, ale często błędny.
2. Numer wiersza dotyczy miejsca, gdzie był kompilator, kiedy wykrył błąd. Sam błąd może znajdować się wcześniej.
3. Po wykryciu błędu kompilator prowizorycznie go „naprawia”, żeby móc iść dalej. To naprawianie może spowodować lawinę innych błędów.
4. Sugestia (jeśli jest) na ogół jest niewłaściwa.

***Bugs* czyli pluskwy**

Przykład: (liczenie średnich — **BŁĘDY** działania)

Testujemy na prostych danych

***Bugs* czyli pluskwy**

Przykład: (liczenie średnich — **BŁĘDY** działania)

Testujemy na prostych danych:

```
Wiecej danych? (Tak/Nie) t
Nowa dana: 2
srednia arytmetyczna == 2.000000
srednia geometryczna == 2.000000
srednia harmoniczna == 2.000000
```

Bugs czyli pluskwy

Przykład: (liczenie średnich — **BŁĘDY** działania)

Testujemy na prostych danych:

```
Wiecej danych? (Tak/Nie) t
```

```
Nowa dana: 2
```

```
srednia arytmetyczna == 2.000000
```

```
srednia geometryczna == 2.000000
```

```
srednia harmoniczna == 2.000000
```


$$\frac{2}{1} = 2$$

Bugs czyli pluskwy

Przykład: (liczenie średnich — **BŁĘDY** działania)

Testujemy na prostych danych:

Wiecej danych? (Tak/Nie) t

Nowa dana: 2

srednia arytmetyczna == 2.000000

srednia geometryczna == 2.000000

srednia harmoniczna == 2.000000

$\frac{2}{1} = 2$ — OK



Bugs czyli pluskwy

Przykład: (liczenie średnich — **BŁĘDY** działania)

Testujemy na prostych danych:

Wiecej danych? (Tak/Nie) t

Nowa dana: 2

srednia arytmetyczna == 2.000000

srednia geometryczna == 2.000000

srednia harmoniczna == 2.000000

$\frac{2}{1} = 2$ — OK

$\sqrt[1]{2} = 2$

Bugs czyli pluskwy

Przykład: (liczenie średnich — **BŁĘDY** działania)

Testujemy na prostych danych:

Wiecej danych? (Tak/Nie) t

Nowa dana: 2

srednia arytmetyczna == 2.000000

srednia geometryczna == 2.000000

srednia harmoniczna == 2.000000

$\frac{2}{1} = 2$ — OK

$\sqrt[1]{2} = 2$ — OK

Bugs czyli pluskwy

Przykład: (liczenie średnich — **BŁĘDY** działania)

Testujemy na prostych danych:

Wiecej danych? (Tak/Nie) t

Nowa dana: 2

srednia arytmetyczna == 2.000000

srednia geometryczna == 2.000000

srednia harmoniczna == 2.000000

$\frac{2}{1} = 2$ — OK

$\sqrt[1]{2} = 2$ — OK

$\frac{1}{\frac{1}{2}} = 2$

Bugs czyli pluskwy

Przykład: (liczenie średnich — **BŁĘDY** działania)

Testujemy na prostych danych:

Wiecej danych? (Tak/Nie) t

Nowa dana: 2

srednia arytmetyczna == 2.000000

srednia geometryczna == 2.000000

srednia harmoniczna == 2.000000


$$\frac{2}{1} = 2 \text{ — OK}$$


$$\sqrt[1]{2} = 2 \text{ — OK}$$


$$\frac{1}{\frac{1}{2}} = 2 \text{ — OK}$$

Bugs czyli pluskwy

Przykład: (liczenie średnich — **BŁĘDY** działania)

Testujemy na prostych danych:

Wiecej danych? (Tak/Nie) t

Nowa dana: 2

srednia arytmetyczna == 2.000000

srednia geometryczna == 2.000000

srednia harmoniczna == 2.000000

$\frac{2}{1} = 2$ — OK

$\sqrt[1]{2} = 2$ — OK

$\frac{1}{\frac{1}{2}} = 2$ — OK

Wiecej danych? (Tak/Nie) t

Nowa dana: 8

srednia arytmetyczna == 5.000000

srednia geometryczna == 1.000000

srednia harmoniczna == 3.200000

Bugs czyli pluskwy

Przykład: (liczenie średnich — **BŁĘDY** działania)

Testujemy na prostych danych:

Wiecej danych? (Tak/Nie) t

Nowa dana: 2

srednia arytmetyczna == 2.000000

$$\frac{2}{1} = 2 \text{ — OK}$$

srednia geometryczna == 2.000000

$$\sqrt[1]{2} = 2 \text{ — OK}$$

srednia harmoniczna == 2.000000

$$\frac{1}{\frac{1}{2}} = 2 \text{ — OK}$$

Wiecej danych? (Tak/Nie) t

Nowa dana: 8

srednia arytmetyczna == 5.000000

$$\frac{2+8}{2} = 5$$

srednia geometryczna == 1.000000

srednia harmoniczna == 3.200000

Bugs czyli pluskwy

Przykład: (liczenie średnich — **BŁĘDY** działania)

Testujemy na prostych danych:

Wiecej danych? (Tak/Nie) t

Nowa dana: 2

srednia arytmetyczna == 2.000000

srednia geometryczna == 2.000000

srednia harmoniczna == 2.000000

$$\frac{2}{1} = 2 \text{ — OK}$$

$$\sqrt[1]{2} = 2 \text{ — OK}$$

$$\frac{1}{\frac{1}{2}} = 2 \text{ — OK}$$

Wiecej danych? (Tak/Nie) t

Nowa dana: 8

srednia arytmetyczna == 5.000000

srednia geometryczna == 1.000000

srednia harmoniczna == 3.200000

$$\frac{2+8}{2} = 5 \text{ — OK}$$

Bugs czyli pluskwy

Przykład: (liczenie średnich — **BŁĘDY** działania)

Testujemy na prostych danych:

Wiecej danych? (Tak/Nie) t

Nowa dana: 2

srednia arytmetyczna == 2.000000

$$\frac{2}{1} = 2 \text{ — OK}$$

srednia geometryczna == 2.000000

$$\sqrt[1]{2} = 2 \text{ — OK}$$

srednia harmoniczna == 2.000000

$$\frac{1}{\frac{1}{2}} = 2 \text{ — OK}$$

Wiecej danych? (Tak/Nie) t

Nowa dana: 8

srednia arytmetyczna == 5.000000

$$\frac{2+8}{2} = 5 \text{ — OK}$$

srednia geometryczna == 1.000000

$$\sqrt[2]{2 \cdot 8} = 4$$

srednia harmoniczna == 3.200000

Bugs czyli pluskwy

Przykład: (liczenie średnich — **BŁĘDY** działania)

Testujemy na prostych danych:

Wiecej danych? (Tak/Nie) t

Nowa dana: 2

srednia arytmetyczna == 2.000000

$$\frac{2}{1} = 2 \text{ — OK}$$

srednia geometryczna == 2.000000

$$\sqrt[1]{2} = 2 \text{ — OK}$$

srednia harmoniczna == 2.000000

$$\frac{1}{\frac{1}{2}} = 2 \text{ — OK}$$

Wiecej danych? (Tak/Nie) t

Nowa dana: 8

srednia arytmetyczna == 5.000000

$$\frac{2+8}{2} = 5 \text{ — OK}$$

srednia geometryczna == 1.000000

$$\sqrt[2]{2 \cdot 8} = 4 \text{ — ŻŁE!}$$

srednia harmoniczna == 3.200000

Bugs czyli pluskwy

Przykład: (liczenie średnich — **BŁĘDY** działania)

Testujemy na prostych danych:

Wiecej danych? (Tak/Nie) t

Nowa dana: 2

srednia arytmetyczna == 2.000000

$$\frac{2}{1} = 2 \text{ — OK}$$

srednia geometryczna == 2.000000

$$\sqrt[1]{2} = 2 \text{ — OK}$$

srednia harmoniczna == 2.000000

$$\frac{1}{\frac{1}{2}} = 2 \text{ — OK}$$

Wiecej danych? (Tak/Nie) t

Nowa dana: 8

srednia arytmetyczna == 5.000000

$$\frac{2+8}{2} = 5 \text{ — OK}$$

srednia geometryczna == 1.000000

$$\sqrt[2]{2 \cdot 8} = 4 \text{ — ŻŁE!}$$

srednia harmoniczna == 3.200000

Średnia geometryczna jest niepoprawnie liczona:

$$\sqrt[2]{2 \cdot 8}$$

Bugs czyli pluskwy

Przykład: (liczenie średnich — **BŁĘDY** działania)

Testujemy na prostych danych:

Wiecej danych? (Tak/Nie) t

Nowa dana: 2

srednia arytmetyczna == 2.000000

$$\frac{2}{1} = 2 \text{ — OK}$$

srednia geometryczna == 2.000000

$$\sqrt[1]{2} = 2 \text{ — OK}$$

srednia harmoniczna == 2.000000

$$\frac{1}{\frac{1}{2}} = 2 \text{ — OK}$$

Wiecej danych? (Tak/Nie) t

Nowa dana: 8

srednia arytmetyczna == 5.000000

$$\frac{2+8}{2} = 5 \text{ — OK}$$

srednia geometryczna == 1.000000

$$\sqrt[2]{2 \cdot 8} = 4 \text{ — ŻŁE!}$$

srednia harmoniczna == 3.200000

Średnia geometryczna jest niepoprawnie liczona:

$$\sqrt[2]{2 \cdot 8} = (2 \cdot 8)^{1/2}$$

Bugs czyli pluskwy

Przykład: (liczenie średnich — **BŁĘDY** działania)

Testujemy na prostych danych:

Wiecej danych? (Tak/Nie) t

Nowa dana: 2

srednia arytmetyczna == 2.000000

$$\frac{2}{1} = 2 \text{ — OK}$$

srednia geometryczna == 2.000000

$$\sqrt[1]{2} = 2 \text{ — OK}$$

srednia harmoniczna == 2.000000

$$\frac{1}{\frac{1}{2}} = 2 \text{ — OK}$$

Wiecej danych? (Tak/Nie) t

Nowa dana: 8

srednia arytmetyczna == 5.000000

$$\frac{2+8}{2} = 5 \text{ — OK}$$

srednia geometryczna == 1.000000

$$\sqrt[2]{2 \cdot 8} = 4 \text{ — ŻŁE!}$$

srednia harmoniczna == 3.200000

Średnia geometryczna jest niepoprawnie liczona:

$$\sqrt[2]{2 \cdot 8} = (2 \cdot 8)^{1/2} = 16^{1/2}$$

Bugs czyli pluskwy

Przykład: (liczenie średnich — **BŁĘDY** działania)

Testujemy na prostych danych:

Wiecej danych? (Tak/Nie) t

Nowa dana: 2

srednia arytmetyczna == 2.000000

$$\frac{2}{1} = 2 \text{ — OK}$$

srednia geometryczna == 2.000000

$$\sqrt[1]{2} = 2 \text{ — OK}$$

srednia harmoniczna == 2.000000

$$\frac{1}{\frac{1}{2}} = 2 \text{ — OK}$$

Wiecej danych? (Tak/Nie) t

Nowa dana: 8

srednia arytmetyczna == 5.000000

$$\frac{2+8}{2} = 5 \text{ — OK}$$

srednia geometryczna == 1.000000

$$\sqrt[2]{2 \cdot 8} = 4 \text{ — ŻŁE!}$$

srednia harmoniczna == 3.200000

Średnia geometryczna jest niepoprawnie liczona:

$$\sqrt[2]{2 \cdot 8} = (2 \cdot 8)^{1/2} = 16^{1/2} \stackrel{?}{=} 1.0$$

Bugs czyli pluskwy

Przykład: (liczenie średnich — **BŁĘDY** działania)

Testujemy na prostych danych:

Wiecej danych? (Tak/Nie) t

Nowa dana: 2

srednia arytmetyczna == 2.000000

$$\frac{2}{1} = 2 \text{ — OK}$$

srednia geometryczna == 2.000000

$$\sqrt[1]{2} = 2 \text{ — OK}$$

srednia harmoniczna == 2.000000

$$\frac{1}{\frac{1}{2}} = 2 \text{ — OK}$$

Wiecej danych? (Tak/Nie) t

Nowa dana: 8

srednia arytmetyczna == 5.000000

$$\frac{2+8}{2} = 5 \text{ — OK}$$

srednia geometryczna == 1.000000

$$\sqrt[2]{2 \cdot 8} = 4 \text{ — ŻŁE!}$$

srednia harmoniczna == 3.200000

Średnia geometryczna jest niepoprawnie liczona:

$$\sqrt[2]{2 \cdot 8} = (2 \cdot 8)^{1/2} = 16^{1/2} \stackrel{?}{=} 1.0$$

$\uparrow$
 $1/2 = 0$ — dzielenie całkowite

Bugs czyli pluskwy

Przykład: (liczenie średnich — **BŁĘDY** działania)

Testujemy na prostych danych:

Wiecej danych? (Tak/Nie) t

Nowa dana: 2

srednia arytmetyczna == 2.000000

$$\frac{2}{1} = 2 \text{ — OK}$$

srednia geometryczna == 2.000000

$$\sqrt[1]{2} = 2 \text{ — OK}$$

srednia harmoniczna == 2.000000

$$\frac{1}{\frac{1}{2}} = 2 \text{ — OK}$$

Wiecej danych? (Tak/Nie) t

Nowa dana: 8

srednia arytmetyczna == 5.000000

$$\frac{2+8}{2} = 5 \text{ — OK}$$

srednia geometryczna == 1.000000

$$\sqrt[2]{2 \cdot 8} = 4 \text{ — ŻŁE!}$$

srednia harmoniczna == 3.200000

Średnia geometryczna jest niepoprawnie liczona:

$$\sqrt[2]{2 \cdot 8} = (2 \cdot 8)^{1/2} = 16^{1/2} \stackrel{?}{=} 1.0$$

↑
 $1/2 = 0$ — dzielenie całkowite
 trzeba zmienić na rzeczywiste: $1.0/2$

Bugs czyli pluskwy

Przykład: (liczenie średnich — **BŁĘDY** działania)

Testujemy na prostych danych:

Wiecej danych? (Tak/Nie) t

Nowa dana: 2

srednia arytmetyczna == 2.000000

$$\frac{2}{1} = 2 \text{ — OK}$$

srednia geometryczna == 2.000000

$$\sqrt[1]{2} = 2 \text{ — OK}$$

srednia harmoniczna == 2.000000

$$\frac{1}{\frac{1}{2}} = 2 \text{ — OK}$$

Wiecej danych? (Tak/Nie) t

Nowa dana: 8

srednia arytmetyczna == 5.000000

$$\frac{2+8}{2} = 5 \text{ — OK}$$

srednia geometryczna == 4.000000

$$\sqrt[2]{2 \cdot 8} = 4 \text{ — OK}$$

srednia harmoniczna == 3.200000

$$\frac{2}{\frac{1}{2} + \frac{1}{8}} = \frac{2}{\frac{5}{8}} = 3.2$$

Bugs czyli pluskwy

Przykład: (liczenie średnich — **BŁĘDY** działania)

Testujemy na prostych danych:

Wiecej danych? (Tak/Nie) t

Nowa dana: 2

srednia arytmetyczna == 2.000000

$$\frac{2}{1} = 2 \text{ — OK}$$

srednia geometryczna == 2.000000

$$\sqrt[1]{2} = 2 \text{ — OK}$$

srednia harmoniczna == 2.000000

$$\frac{1}{\frac{1}{2}} = 2 \text{ — OK}$$

Wiecej danych? (Tak/Nie) t

Nowa dana: 8

srednia arytmetyczna == 5.000000

$$\frac{2+8}{2} = 5 \text{ — OK}$$

srednia geometryczna == 4.000000

$$\sqrt[2]{2 \cdot 8} = 4 \text{ — OK}$$

srednia harmoniczna == 3.200000

$$\frac{2}{\frac{1}{2} + \frac{1}{8}} = \frac{2}{\frac{5}{8}} = 3.2 \text{ — OK}$$

Bugs czyli pluskwy

Przykład: (liczenie średnich — **BŁĘDY** działania)

Testujemy na prostych danych:

Wiecej danych? (Tak/Nie) t

Nowa dana: 2

srednia arytmetyczna == 2.000000

$$\frac{2}{1} = 2 \text{ — OK}$$

srednia geometryczna == 2.000000

$$\sqrt[1]{2} = 2 \text{ — OK}$$

srednia harmoniczna == 2.000000

$$\frac{1}{\frac{1}{2}} = 2 \text{ — OK}$$

Wiecej danych? (Tak/Nie) t

Nowa dana: 8

srednia arytmetyczna == 5.000000

$$\frac{2+8}{2} = 5 \text{ — OK}$$

srednia geometryczna == 4.000000

$$\sqrt[2]{2 \cdot 8} = 4 \text{ — OK}$$

srednia harmoniczna == 3.200000

$$\frac{2}{\frac{1}{2} + \frac{1}{8}} = \frac{2}{\frac{5}{8}} = 3.2 \text{ — OK}$$

Wiecej danych? (Tak/Nie) t

Nowa dana: 0

srednia arytmetyczna == 3.333333

srednia geometryczna == 0.000000

srednia harmoniczna == 0.000000

Bugs czyli pluskwy

Przykład: (liczenie średnich — **BŁĘDY** działania)

Testujemy na prostych danych:

Wiecej danych? (Tak/Nie) t

Nowa dana: 2

srednia arytmetyczna == 2.000000

$$\frac{2}{1} = 2 \text{ — OK}$$

srednia geometryczna == 2.000000

$$\sqrt[1]{2} = 2 \text{ — OK}$$

srednia harmoniczna == 2.000000

$$\frac{1}{\frac{1}{2}} = 2 \text{ — OK}$$

Wiecej danych? (Tak/Nie) t

Nowa dana: 8

srednia arytmetyczna == 5.000000

$$\frac{2+8}{2} = 5 \text{ — OK}$$

srednia geometryczna == 4.000000

$$\sqrt[2]{2 \cdot 8} = 4 \text{ — OK}$$

srednia harmoniczna == 3.200000

$$\frac{2}{\frac{1}{2} + \frac{1}{8}} = \frac{2}{\frac{5}{8}} = 3.2 \text{ — OK}$$

Wiecej danych? (Tak/Nie) t

Nowa dana: 0

srednia arytmetyczna == 3.333333

$$\frac{2+8+0}{3} = 3.33 \dots$$

srednia geometryczna == 0.000000

srednia harmoniczna == 0.000000

Bugs czyli pluskwy

Przykład: (liczenie średnich — **BŁĘDY** działania)

Testujemy na prostych danych:

Wiecej danych? (Tak/Nie) t

Nowa dana: 2

srednia arytmetyczna == 2.000000

$$\frac{2}{1} = 2 \text{ — OK}$$

srednia geometryczna == 2.000000

$$\sqrt[1]{2} = 2 \text{ — OK}$$

srednia harmoniczna == 2.000000

$$\frac{1}{\frac{1}{2}} = 2 \text{ — OK}$$

Wiecej danych? (Tak/Nie) t

Nowa dana: 8

srednia arytmetyczna == 5.000000

$$\frac{2+8}{2} = 5 \text{ — OK}$$

srednia geometryczna == 4.000000

$$\sqrt[2]{2 \cdot 8} = 4 \text{ — OK}$$

srednia harmoniczna == 3.200000

$$\frac{2}{\frac{1}{2} + \frac{1}{8}} = \frac{2}{\frac{5}{8}} = 3.2 \text{ — OK}$$

Wiecej danych? (Tak/Nie) t

Nowa dana: 0

srednia arytmetyczna == 3.333333

$$\frac{2+8+0}{3} = 3.33... \text{ — OK}$$

srednia geometryczna == 0.000000

srednia harmoniczna == 0.000000

Bugs czyli pluskwy

Przykład: (liczenie średnich — **BŁĘDY** działania)

Testujemy na prostych danych:

Wiecej danych? (Tak/Nie) t

Nowa dana: 2

srednia arytmetyczna == 2.000000

$$\frac{2}{1} = 2 \text{ — OK}$$

srednia geometryczna == 2.000000

$$\sqrt[1]{2} = 2 \text{ — OK}$$

srednia harmoniczna == 2.000000

$$\frac{1}{\frac{1}{2}} = 2 \text{ — OK}$$

Wiecej danych? (Tak/Nie) t

Nowa dana: 8

srednia arytmetyczna == 5.000000

$$\frac{2+8}{2} = 5 \text{ — OK}$$

srednia geometryczna == 4.000000

$$\sqrt[2]{2 \cdot 8} = 4 \text{ — OK}$$

srednia harmoniczna == 3.200000

$$\frac{2}{\frac{1}{2} + \frac{1}{8}} = \frac{2}{\frac{5}{8}} = 3.2 \text{ — OK}$$

Wiecej danych? (Tak/Nie) t

Nowa dana: 0

srednia arytmetyczna == 3.333333

$$\frac{2+8+0}{3} = 3.33... \text{ — OK}$$

srednia geometryczna == 0.000000

$$\sqrt[3]{2 \cdot 8 \cdot 0} = 0$$

srednia harmoniczna == 0.000000

Bugs czyli pluskwy

Przykład: (liczenie średnich — **BŁĘDY** działania)

Testujemy na prostych danych:

Wiecej danych? (Tak/Nie) t

Nowa dana: 2

srednia arytmetyczna == 2.000000

$$\frac{2}{1} = 2 \text{ — OK}$$

srednia geometryczna == 2.000000

$$\sqrt[1]{2} = 2 \text{ — OK}$$

srednia harmoniczna == 2.000000

$$\frac{1}{\frac{1}{2}} = 2 \text{ — OK}$$

Wiecej danych? (Tak/Nie) t

Nowa dana: 8

srednia arytmetyczna == 5.000000

$$\frac{2+8}{2} = 5 \text{ — OK}$$

srednia geometryczna == 4.000000

$$\sqrt[2]{2 \cdot 8} = 4 \text{ — OK}$$

srednia harmoniczna == 3.200000

$$\frac{2}{\frac{1}{2} + \frac{1}{8}} = \frac{2}{\frac{5}{8}} = 3.2 \text{ — OK}$$

Wiecej danych? (Tak/Nie) t

Nowa dana: 0

srednia arytmetyczna == 3.333333

$$\frac{2+8+0}{3} = 3.33... \text{ — OK}$$

srednia geometryczna == 0.000000

$$\sqrt[3]{2 \cdot 8 \cdot 0} = 0 \text{ — OK}$$

srednia harmoniczna == 0.000000

Bugs czyli pluskwy

Przykład: (liczenie średnich — **BŁĘDY** działania)

Testujemy na prostych danych:

Wiecej danych? (Tak/Nie) t

Nowa dana: 2

srednia arytmetyczna == 2.000000

$$\frac{2}{1} = 2 \text{ — OK}$$

srednia geometryczna == 2.000000

$$\sqrt[1]{2} = 2 \text{ — OK}$$

srednia harmoniczna == 2.000000

$$\frac{1}{\frac{1}{2}} = 2 \text{ — OK}$$

Wiecej danych? (Tak/Nie) t

Nowa dana: 8

srednia arytmetyczna == 5.000000

$$\frac{2+8}{2} = 5 \text{ — OK}$$

srednia geometryczna == 4.000000

$$\sqrt[2]{2 \cdot 8} = 4 \text{ — OK}$$

srednia harmoniczna == 3.200000

$$\frac{2}{\frac{1}{2} + \frac{1}{8}} = \frac{2}{\frac{5}{8}} = 3.2 \text{ — OK}$$

Wiecej danych? (Tak/Nie) t

Nowa dana: 0

srednia arytmetyczna == 3.333333

$$\frac{2+8+0}{3} = 3.33... \text{ — OK}$$

srednia geometryczna == 0.000000

$$\sqrt[3]{2 \cdot 8 \cdot 0} = 0 \text{ — OK}$$

srednia harmoniczna == 0.000000

$$\frac{3}{\frac{1}{2} + \frac{1}{8} + \frac{1}{0}}$$

Bugs czyli pluskwy

Przykład: (liczenie średnich — **BŁĘDY** działania)

Testujemy na prostych danych:

Wiecej danych? (Tak/Nie) t

Nowa dana: 2

srednia arytmetyczna == 2.000000

srednia geometryczna == 2.000000

srednia harmoniczna == 2.000000

$$\frac{2}{1} = 2 \text{ — OK}$$

$$\sqrt[1]{2} = 2 \text{ — OK}$$

$$\frac{1}{\frac{1}{2}} = 2 \text{ — OK}$$

Wiecej danych? (Tak/Nie) t

Nowa dana: 8

srednia arytmetyczna == 5.000000

srednia geometryczna == 4.000000

srednia harmoniczna == 3.200000

$$\frac{2+8}{2} = 5 \text{ — OK}$$

$$\sqrt[2]{2 \cdot 8} = 4 \text{ — OK}$$

$$\frac{2}{\frac{1}{2} + \frac{1}{8}} = \frac{2}{\frac{5}{8}} = 3.2 \text{ — OK}$$

Wiecej danych? (Tak/Nie) t

Nowa dana: 0

srednia arytmetyczna == 3.333333

srednia geometryczna == 0.000000

srednia harmoniczna == 0.000000

$$\frac{2+8+0}{3} = 3.33... \text{ — OK}$$

$$\sqrt[3]{2 \cdot 8 \cdot 0} = 0 \text{ — OK}$$

$$\frac{3}{\frac{1}{2} + \frac{1}{8} + \frac{1}{0}} \text{ — dzielenie przez 0!}$$

***Bugs* czyli pluskwy**

... to jeszcze nie koniec...

***Bugs* czyli pluskwy**

... to jeszcze nie koniec...

Dla niektórych danych obliczenie nie ma sensu

***Bugs* czyli pluskwy**

... to jeszcze nie koniec...

Dla niektórych danych obliczenie nie ma sensu — program powinien podawać przyczynę i **nie liczyć**.

***Bugs* czyli pluskwy**

... to jeszcze nie koniec...

Dla niektórych danych obliczenie nie ma sensu — program powinien podawać przyczynę i **nie liczyć**.

Program musi zostać **starannie przetestowany**, zanim zostanie uznany za gotowy.

***Bugs* czyli pluskwy**

... to jeszcze nie koniec...

Dla niektórych danych obliczenie nie ma sensu — program powinien podawać przyczynę i **nie liczyć**.

Program musi zostać **starannie przetestowany**, zanim zostanie uznany za gotowy.

Kiedy długi program działa błędnie, to znalezienie błędu może być trudne.

***Bugs* czyli pluskwy**

... to jeszcze nie koniec...

Dla niektórych danych obliczenie nie ma sensu — program powinien podawać przyczynę i **nie liczyć**.

Program musi zostać **starannie przetestowany**, zanim zostanie uznany za gotowy.

Kiedy długi program działa błędnie, to znalezienie błędu może być trudne. Często wtedy pomaga tymczasowe wmontowanie w niego **próbnych wydruków**

***Bugs* czyli pluskwy**

... to jeszcze nie koniec...

Dla niektórych danych obliczenie nie ma sensu — program powinien podawać przyczynę i **nie liczyć**.

Program musi zostać **starannie przetestowany**, zanim zostanie uznany za gotowy.

Kiedy długi program działa błędnie, to znalezienie błędu może być trudne. Często wtedy pomaga tymczasowe wmontowanie w niego **próbnych wydruków** — do usunięcia po wyleczeniu problemu.

Bugs czyli pluskwy

... to jeszcze nie koniec...

Dla niektórych danych obliczenie nie ma sensu — program powinien podawać przyczynę i **nie liczyć**.

Program musi zostać **starannie przetestowany**, zanim zostanie uznany za gotowy.

Kiedy długi program działa błędnie, to znalezienie błędu może być trudne. Często wtedy pomaga tymczasowe wmontowanie w niego **próbnych wydruków** — do usunięcia po wyleczeniu problemu.

Rodzaje błędów:

łatwiejsze	leksykalne	— źle napisane słowo; np. duble
	składniowe	— źle zestawione elementy; np. pomyłone nawiasy
trudniejsze	pętla	— program liczy bez końca
	logiczne	— program kończy obliczenie z niepoprawnym wynikiem

Bugs czyli pluskwy

... to jeszcze nie koniec...

Dla niektórych danych obliczenie nie ma sensu — program powinien podawać przyczynę i **nie liczyć**.

Program musi zostać **starannie przetestowany**, zanim zostanie uznany za gotowy.

Kiedy długi program działa błędnie, to znalezienie błędu może być trudne. Często wtedy pomaga tymczasowe wmontowanie w niego **próbnych wydruków** — do usunięcia po wyleczeniu problemu.

Rodzaje błędów:

<i>łatwiejsze</i>	leksykalne składniowe	— źle napisane słowo; np. duble — źle zestawione elementy; np. pomyłone nawiasy	} ← wykrywane przy kompilacji
<i>trudniejsze</i>	pętla logiczne	— program liczy bez końca — program kończy obliczenie z niepoprawnym wynikiem	