

## Podstawy programowania, Czwartek 16.05.2015, 8-10

### Zadanie 9

Zadanie polega na napisaniu programu służącego do obliczania miejsc zerowych wielomianu stopnia drugiego o zadanych parametrach. W programie tworzymy odpowiednią strukturę `Wielomian`, której podajemy parametry wielomianu a następnie obliczamy liczbę miejsc zerowych, wartości miejsc zerowych, oraz wypisujemy wynik obliczeń na ekran.

Struktura `Wielomian` powinna mieć następujące składowe: `double a`, `double b`, `double c`, `int n`, `double *x`, gdzie 3 pierwsze parametry (`a`, `b`, `c`) to parametry wielomianu:

$$W(x) = ax^2 + bx + c$$

`n` oznacza liczbę miejsc zerowych, a `x` tablicę miejsc zerowych (alokowaną dynamicznie, czyli o rozmiarze 1 lub 2, w zależności od `n`).

Oprócz struktury powinny być stworzone również funkcje: `void UstawWielomian(Wielomian *w, double A, double B, double C)`, `void WypiszWielomian(Wielomian *w)`. Należy również stworzyć funkcje pomocnicze: `double Delta(Wielomian *w)`, `int LiczbaMiejscZerowych(Wielomian *w)`, `void MiejscaZerowe(Wielomian *w)`, które posłużą do obliczania liczby miejsc zerowych oraz samych pierwiastków równania kwadratowego. Uwaga, wywołanie funkcji `UstawWielomian` z argumentami `A`, `B`, `C` ustawionymi na 0 powoduje wylosowanie (używamy funkcji `rand()`) parametrów `a`, `b`, `c` wielomianu z zakresu  $-10 \dots 10$  (losujemy wartości rzeczywiste).

Punktacja:

1. Stworzenie wielomianu `w1` (parametry: 1, -8, 12) oraz wielomianu `w2` (parametry ustawiane losowo) (1.5 p.)
2. Obliczenie liczby miejsc zerowych (1 p.)
3. Obliczenie miejsc zerowych (w tym odpowiednia alokacja tablicy) (2 p.)
4. Wypisanie obliczonych parametrów wielomianu (1 p.)

Całość kompilujemy w linii poleceń używając pliku `Makefile` oraz polecenia `make`. (0.5 p.)

### Uwaga!

W plikach `.h` (gdzie mamy definicje struktur) używamy następujących instrukcji preprocesora:

```
#ifndef WIELOMIAN_H
#define WIELOMIAN_H
... (zawartość pliku)
#endif //WIELOMIAN_H
```

## Uwaga 2!

### Przykład piku Makefile

CC=gcc

CFLAGS=-Wall -pedantic -std=c99

LIBS=-lm

all: program

program: main.o struktura1.o struktura2.o

\$(CC) -o main main.o struktura1.o struktura2.o \$(CFLAGS) \$(LIBS)

struktura1.o: struktura1.c struktura1.h struktura2.h

\$(CC) -o struktura1.o -c struktura1.c \$(CFLAGS)

struktura2.o: struktura2.c struktura2.h struktura1.h

\$(CC) -o struktura2.o -c struktura2.c \$(CFLAGS)

main.o: main.c struktura1.h struktura2.h

\$(CC) -o main.o -c main.c \$(CFLAGS)

clean:

rm -rf \*.o main