

# Podstawy Programowania: Przecięcia zera"

Dawid Krekora

18 listopada 2020

# 1 Opis programu

Program którego zaimplementowanie było zadaniem przedstawionym na laboratoriach miał być w założeniu symulatorem urządzenia badającego puls człowieka. Głównym zadaniem programu jest pobieranie danych ze standardowego strumienia wejścia, a konkretnie sekwencji składającej się z 99 liczb rzeczywistych. Dwie sąsiednie liczby są ze sobą porównywane w celu sprawdzenia czy pomiędzy nimi wystąpiło przecięcie zera - czyli sytuacja która nas interesuje i która później na tej podstawie pozwoli wyliczyć tętno człowieka.

# 2 Kod programu

```
1
2 #include <stdio.h>
3
4 #define MIN_VALUE -10 //minimalna wartosc liczb wliczanych do zakresu
5 #define MAX_VALUE 10 //maksymalna wartosc liczb wliczanych do zakresu
6 #define MIN_FREQ 8 //minimalna, prawidlowa czestotliwosc pulsu
7 #define MAX_FREQ 14 //maksymalna, prawidlowa czestotliwosc pulsu
8 #define CRIT_VALUE 99 //liczba oznaczajaca koniec wczytywania ze strumienia danych
9 #define DATA_VALUE 99 //maksymalna liczba pobranych danych w jednej sekwencji
10
11 int main ()
12 {
13     //deklaracja zmiennych i nadanie im wartosci poczatkowych:
14     float a = 0.0; //zmienna do ktorej wczytujemy dane wejsciowe
15     float b = 0.0; //zmienna pomocnicza do przechowywania poprzedniej wartosci
16     int cross_0 = 0; //licznik wystapienia przeciecia zera
17     int counter = 0; //licznik wczytanych liczb w sekwencji
18
19     while (a != CRIT_VALUE) { //petla sprawdzajaca kiedy zakonczyc dzialanie programu
20         scanf("%f", &a);
21         counter++;
22
23         if (a>MIN_VALUE && a<MAX_VALUE && a!=0) { //instrukcja warunkowa
24             sprawdzajaca czy liczba miesci sie w poprawnym zakresie
25             if (a*b < 0){
26                 cross_0++;
27             }
28             b = a;
29             if (!(counter % DATA_VALUE) || b==CRIT_VALUE) { //instrukcja warunkowa sprawdzajaca
30                 kiedy zakonczyc pojedyncza sekwencje liczb i przystapic do interpretacji wynikow
31                 if (cross_0<MIN_FREQ || cross_0>MAX_FREQ){ //instrukcja warunkowa interpretujaca
32                     wartosc tetna
33                     printf("Niepoprawny puls (%d)\n",cross_0);
34                 }
35                 else {
36                     printf("Puls w normie. Wartosc pulsu: %d\n", cross_0);
37                 }
38                 a = b = 0.0; //wyzerowanie wartosci poczatkowych
39                 zmiennych dla kolejnej sekwencji liczb
40                 cross_0 = counter = 0;
41             }
42         }
43     }
```

```

38 }
39
40 return 0;
41 }
42
43

```

Listing 1: Kod programu

### 3 Wprowadzanie danych do programu oraz kompilacja. Testy na przygotowanych danych.

Kompletny program potrzebuje danych wejściowych które byłyby wprowadzane z klawiatury (niezalecane) lub wpisane do pliku ,a następnie przekierowane na standardowy strumień wejściowy. Jedno z zadań związanych z laboratorium była prawidłowa konwersja wartości zawartych w plikach tekstowych do formy która byłaby zrozumiała przez program. W tym celu użyto komend które pozbywają się niepotrzebnych wartości które przedstawiają czas oraz późniejszych białych znaków. "Wyłuskane"dane zostały przekierowane i zapisane w osobnym pliku tekstowym, gotowym do późniejszego użycia (przekierowania do programu)(Rys.1). Wynik działania programu został przedstawiony na rysunku nr.2.

```

dawid@dawid-Aspire-E5-575T:~/Pulpit$ cut -d ' ' -f 2 dane1.txt | cut -d',' -f1 > dane1_finalne.txt
dawid@dawid-Aspire-E5-575T:~/Pulpit$ cut -d ' ' -f 2- dane2.txt | cut -d',' -f1 > dane2_finalne.txt
dawid@dawid-Aspire-E5-575T:~/Pulpit$

```

Rysunek 1: Schemat blokowy algorytmu do równań kwadratowych

```

dawid@dawid-Aspire-E5-575T:~/Pulpit$ gcc -o zera przeciecia_zera.c
dawid@dawid-Aspire-E5-575T:~/Pulpit$ ./zera < dane1_finalne.txt
Puls w normie. Wartosc pulsu: 11
Niepoprawny puls (5)
Puls w normie. Wartosc pulsu: 10
Puls w normie. Wartosc pulsu: 12
Puls w normie. Wartosc pulsu: 12
Puls w normie. Wartosc pulsu: 12
Puls w normie. Wartosc pulsu: 12
Puls w normie. Wartosc pulsu: 13
Niepoprawny puls (18)
Puls w normie. Wartosc pulsu: 11
dawid@dawid-Aspire-E5-575T:~/Pulpit$ ./zera < dane2_finalne.txt
Puls w normie. Wartosc pulsu: 11
Niepoprawny puls (5)
Puls w normie. Wartosc pulsu: 10
Puls w normie. Wartosc pulsu: 12
Puls w normie. Wartosc pulsu: 12
Puls w normie. Wartosc pulsu: 12
Puls w normie. Wartosc pulsu: 12
Puls w normie. Wartosc pulsu: 13
Niepoprawny puls (18)
Puls w normie. Wartosc pulsu: 11

```

Rysunek 2: Schemat blokowy algorytmu do równań kwadratowych

Po skonsultowaniu działania kodu z innymi, niezależnymi rezultatami kodu innych osób dowiedziono jego poprawnego działania.

## 4 Testy na mniejszych, przygotowanych paczkach.

W celu sprawdzenia poprawności działania programu przeprowadzono testy na mniejszych pakietach danych w taki sposób aby rezultat działań dało się przewidzieć nie stosując skomplikowanych obliczeń. Zmniejszono liczbę wartości w sekwencji do 5 liczb oraz zmniejszono przedział oznaczający prawidłowy puls z 14 na 4 oraz z 8 na 1.

### 4.1 Zwykłe wartości

- Dane wejściowe: -1.0 1.0 -1.0 1.0 -1.0
- Oczekiwany rezultat: Wartość pulsu: 4
- Rezultat: Wartość pulsu: 4
- Test: pozytywny

### 4.2 Sama liczba 99

- Dane wejściowe: 99
- Oczekiwany rezultat: koniec działania programu
- Rezultat: koniec działania programu
- Test: pozytywny

### 4.3 Zera pośród danych

- Dane wejściowe: -1.0 0.0 1.0 0.0 0.0
- Oczekiwany rezultat: brak reakcji programu
- Rezultat: program nie skończył swojego działania ponieważ pomija zera, a więc według niego wpisano tylko 2 wartości (do zakończenia działania sekwencji potrzeba ich 5)
- Test: pozytywny

### 4.4 Wartości poza zakresem

- Dane wejściowe: -1.0 11.0 2.0 -20.0 -4.0
- Oczekiwany rezultat: Wartość pulsu: 2
- Rezultat: Wartość pulsu: 2
- Test: pozytywny

## 5 Podsumowanie i wnioski

Program został przetestowany tylko przy pomocy danych liczbowych zdając test pozytywnie. Nie został jednak przystosowany do radzenia sobie z błędami wynikającymi ze złego sposobu wprowadzenia danych czy też działania na symbolach nieliczbowych. W pozostałych kwestiach działa zgodnie z ideą jego stworzenia i z powodzeniem mógłby poradzić sobie z zadaniem związanym z liczeniem pulsu (przy pomocy odpowiedniej aparatury która dostarczyłaby dane).