

Programowanie proceduralne lab 8

Zadanie 1. (2 pkt)

Przeanalizuj przykład ptrfn.c, rozbuduj go o funkcję liczącą x^{-2} , w analogiczny sposób dołącz wskaźnik na swoją funkcję do tablicy wskaźników i wykonaj obliczenia, dla dziesięciu dowolnie zdefiniowanych argumentów funkcji (np. dla dziesięciu kolejnych liczb parzystych), wypisując dla każdego z argumentów rezultaty wykonania wszystkich, dostępnych poprzez zawarte w tablicy wskaźniki, funkcji.

Zadanie 2. (2 pkt)

Zdefiniuj funkcję, która zwraca sumą dwóch liczb całkowitych, oraz funkcję, która zwraca wartość większą z dwóch liczb całkowitych. Niech prototypy funkcji mają postać:

```
int add2(int *, int *)
int comp2(int *, int *)
```

zdefiniuj funkcję:

```
int add2_or_comp2 (),
```

której argumenty to:

dwie wartości całkowite,

wskaźnik do funkcji, która pobiera dwa wskaźniki na int, a zwraca wartość typu int

funkcja ma zwracać wartość typu int, która jest wynikiem przekazanej przez wskaźnik funkcji.

W funkcji main() pobierz dwie wartości całkowite wywołaj funkcje `add2_or_comp2()` podając jako argument wskaźnik do funkcji `add2()` i wypisz ich sumę, następnie wypisz wyniki porównania tych wartości analogicznie wywołując funkcję podając jako argument wskaźnik do funkcji `comp2()`.

Zadanie 2. (3 pkt)

Napisz i przetestuj dwie funkcje wyliczające $x!$ dla zadanego parametru.

Funkcje realizują obliczenia na dwa sposoby iteracyjnie oraz rekurencyjnie.

Program liczący $x!$ uzupełnij o możliwość zadania w linii poleceń parametru x oraz sposobu realizacji obliczeń (rekurencja/iteracja). Przykładowe wywołanie / a.out 12 rekurencja

W przypadku braku argumentów w linii poleceń uruchamiania programu proszę wypisać stosowny komunikat jak powinno wyglądać uruchomienie programu.

Użyj wskaźnika do funkcji aby wywołać jedną z metod liczenia silni w zależności od wyboru użytkownika.

Wydruk z realizacji rekurencyjnej wersji obliczeń powinien wyglądać następująco:

```
wartosc k =5, adres k= 0x7ffcd85c118c
. wartosc k =4, adres k= 0x7ffcd85c115c
.. wartosc k =3, adres k= 0x7ffcd85c112c
... wartosc k =2, adres k= 0x7ffcd85c10fc
.... wartosc k =1, adres k= 0x7ffcd85c10cc
... wartosc k =2, adres k = 0x7ffcd85c10fc wartosc wyn = 2, adres wyn=0x7ffcd85c1108
.. wartosc k =3, adres k = 0x7ffcd85c112c wartosc wyn = 6, adres wyn=0x7ffcd85c1138
. wartosc k =4, adres k = 0x7ffcd85c115c wartosc wyn = 24, adres wyn=0x7ffcd85c1168
wartosc k =5, adres k = 0x7ffcd85c118c wartosc wyn = 120, adres wyn=0x7ffcd85c1198
```

Zadanie 4. (3 pkt)

Utwórz tablicę zawierającą 1000 losowych liczb całkowitych, z zakresu (-100 ;100) posortuj używając funkcji `qsort()`. Sprawdź za pomocą funkcji `bsearch()` czy zadane jako argumenty programu wartości znajdują się w tablicy oraz określ jakie są wielokrotności ich występowania.