

## Programowanie proceduralne lab 10\_L3

### Zadanie 1. (3 pkt)

Utwórz strukturę `struct Zespolona{ }`, która posłuży do przechowywania liczb zespolonych.

Utwórz funkcję `zesp_wypisz()`, typu `void`, która przyjmuje jako argument `struct Zespolona` i pozwoli wypisać liczbę zespoloną w postaci :

`re + im*i` (czyli np. `5.3+15.0*i`)

a) utwórz strukturę `Zesp_modul`, która ma 3 pola: zmienną zespoloną (struktura), moduł liczby zespolonej oraz wskaźnik do funkcji typu `void`, która przyjmuje jako argument strukturę `Zespolona`.

Proszę zaalokuj pamięć dla jednowymiarowej tablicy struktur o rozmiarze podanym przez użytkownika, wypełnij liczbami losowymi pola struktury `Zespolona`. Ustaw wskaźnik do funkcji `zesp_wypisz`. Napisz funkcję typu `void`, która wypełni pole moduł każdego z elementów tablicy struktur. Wypisz tablicę na ekran używając zdefiniowanego wskaźnika i zapisz do pliku tekstowego w postaci:

```
| (5.000000+8.000000*i) | =9.433981
| (4.000000+8.000000*i) | =8.944272
| (3.000000+11.000000*i) | =11.401754
| (6.000000+9.000000*i) | =10.816654
| (5.000000+10.000000*i) | =11.180340
| (7.000000+12.000000*i) | =13.892444
```

b) Korzystając z funkcji bibliotecznej `qsort()` posortuj tablicę struktur niemalejąco względem modułów. Wypisz wynik na ekran (używając wskaźnika do funkcji wypisującej liczbę zespoloną) i zapisz do pliku tekstowego.

### Zadanie 2. (4 pkt)

Napisz proszę program, który oblicza wartości zdefiniowanych w nim funkcji arytmetycznych dla zadanych wartości argumentu i wypisuje maksymalną wartość oraz nazwę funkcji, dla której ta maksymalna wartość została znaleziona.

Do realizacji zadania zastosuj poniższy schemat postępowania:

Proszę zdefiniować pięć funkcji :

```
double fun0(double x) { return log(x); }
double fun1(double x) { return x*x; }
double fun2(double x) { return exp(x); }
double fun3(double x) { return sin(x); }
```

Używając `typedef` zdefiniuj typ `WSKFUN_D` - wskaźnik do funkcji, która przyjmuje wartość typu `double` i zwraca `double` (`double → double`). Przy użyciu tego wskaźnika zdefiniuj typ `TABWSKFUN_D` - tablica wskaźników do funkcji `double → double`.

Zdefiniuj typ `TABFUN_NAME` – tablica łańcuchów znaków.

Zdefiniuj funkcję `funmax()`, której pierwszym argumentem będzie tablica wskaźników funkcyjnych, funkcja zwróci wskaźnik do funkcji, dla której wartość obliczona dla drugiego argumentu (zadanej wartości `double x`) jest największa.

W funkcji `main()` utwórz tablicę wskaźników funkcyjnych (typu `TABWSKFUN_D`) i zainicjalizuj wskaźnikami do utworzonych funkcji.

Utwórz tablicę łańcuchów znaków, zawierającą nazwy zdefiniowanych funkcji fun0,...,fun4 - {"log","kwadrat",....}

Utwórz zmienną fmax typu WSKFUN\_D, której przypiszesz wskaźnik do funkcji zwracającej największą wartość dla danego argumentu.

Przetestuj działanie programu dla argumentów z przedziału od wartości  $x=1$  do wartości  $x=10$  z krokiem równym  $k=1$  i wypisz wyniki np. w postaci:

*Dla argumentu  $x=1.0$  największą wartość równą 2.71828 przyjmuje funkcja  $\exp(x)$*

Rozważ przypadki równych wartości max dla kilku funkcji.

### **Zadanie 3. (3 pkt)**

Napisz program który wykonuje dodawanie i odejmowanie liczb całkowitych. Maksymalna długość wprowadzanych operandów może wynosić 200 cyfr.

W funkcji main() wykonaj pobranie operandów i wypisanie wyników, wykonanie obliczeń wykonaj w utworzonych funkcjach