Programowanie Zaawansowane FM i NI (C++)
Ćwiczenia 4

Zadanie 1 (referencje)

(a) Napisz funkcję `DwieSumy`, która dla przedziału liczb całkowitych [a, b], podawanego jej jako argumenty wejścia, oblicza sumę liczb parzystych i nieparzystych. Niech funkcja „zwraca" oba wyniki poprzez mechanizm referencji.

W funkcji `main` wczytaj od użytkownika z klawiatury dwie liczby do zmiennych. Następnie wywołaj funkcję `DwieSumy`, podając jej te zmienne w argumentach wejścia. Na koniec, w funkcji `main` wyświetl wyniki.

(b) Napisz funkcję `void ord3 (double& a, double& b, double& c);` która pobiera trzy liczby typu `double` przez referencje, i porządkuje je w kolejności wzrastającej (tak, że po wyjściu z funkcji ich wartości są zmienione).

W funkcji `main` wczytaj z klawiatury trzy liczby, a następnie wywołaj funkcję `ord3` i wypisz wyniki na ekran.

Zadanie 2 (referencje, algorytm)

Napisz funkcję `Analiza`, która dla funkcji `double f (double x)`, zwracającej wynik wymyślonego przez Ciebie wielomianu lub funkcji trygonometrycznej, przeprowadzi jej prostą analizę w przedziale `x ∈ [a, b]` , iterując `x` co krok `dx` (niech `a, b, dx` będą zmiennymi w argumentach wejścia funkcji `Analiza`.)

Funkcja `Analiza` powinna obliczyć wartość najmniejszą i największą zarówno dla *f*(*x*), jak i jej ilorazu różnicowego, będącego przybliżeniem pochodnej:

$$D(x) = \frac{f(x) - f(x-\Delta x)}{\Delta x}$$

gdzie Δ*x* – jako szerokość kroku – utożsamiamy ze zmienną `dx`.

Wyznaczone wartości `Analiza` powinna zwracać do `main` poprzez mechanizm referencji. W funkcji `main` wypisz wyniki na ekran.

Zadanie 3 (typ char, tabela ascii)

Komputer wymyśla losowo pewną literę (małą). Zadaniem użytkownika jest zgadnięcie jej i ma na to 5 prób. Niech kod w każdym kroku pętli: poprosi użytkownika o zgadnięcie litery, przyjmie ją z klawiatury do zmiennej i odpisze, że litera jest zgadnięta / alfabetycznie za wczesna (za późna) / znak jest błędny.

Jeśli po próbach nie uda się zgadnąć, niech kod napisze komunikat.

Zadanie 4 (kodowanie bitowe)

Przeprowadzamy ankietę na pewien temat. Wynik pojedynczej ankiety (uzyskanej od jednego respondenta) zawiera następujące informacje, które należy zakodować w jednej zmiennej typu `unsigned short` (ma ona 2 bajty, czyli 16 bitów) :

1. płeć – **1 bit** , bo 2 możliwości (kobieta, mężczyzna), kodowane jako **0 lub 1**;
2. stan cywilny – **2 bity**, bo 4 możliwości (panna/kawaler, matka/żonaty, rozwódka/rozwodnik, wdowa/wdowiec) kodowane jako **0, 1, 2 lub 3**;
3. grupa wiekowa – **2 bity**, bo 4 możliwości (np. 18-30, 31-45, 46-60, 60+) kodowane jako **0, 1, 2 lub 3**;
4. wykształcenie – **2 bity**, bo 4 możliwości (np. podstawowe, średnie, licencjat, magister+) kodowane jako **0, 1, 2 lub 3**;
5. miejsce zamieszkania – **2 bity**, bo 4 możliwości (np. wieś, miasto do 50 tys., miasto 50 - 400 tys., miasto ponad 400 tysięcy mieszkańców) kodowane jako **0, 1, 2 lub 3**;
6. region kraju – **4 bity**, bo (przypuśćmy) jest 16 regionów ponumerowanych **od 0 do 15**;
7. odpowiedź na pytanie ankietera – 3 bity, bo w ankiecie (przypuśćmy) było 8 możliwych odpowiedzi, ponumerowanych **od 0 do 7**.

Napisz funkcję:

```
unsigned short koduj (int plec, int stan_cyw, int grupa_wiek,
                      int edu, int zam, int region, int odp );
```

która pobiera 7 liczb (jak wyżej), koduje informacje w nich zawarte w jednej zmiennej typu `unsigned short` i zwraca tę jedną liczbę do funkcji wywołującej. Następnie napisz funkcję:

```
void info (unsigned short kod);
```

która pobiera jeden argument typu `unsigned short`, zawierający zakodowaną informacje o jednej ankiecie, a następnie rozpakowuje i czytelnie wypisuje składniki, np. w formie:

```
plec:           0
stan cywilny:   3
grupa wiekowa:  2
wyksztalcenie:  3
miejsce zam.:   0
region:         12
odpowiedz:      6
```

Do kodowania użyj standardowych operatorów bitowych (nie używaj dedykowanych bibliotek).

Zadanie 5 (wskaźniki)

a) Przepisz kod do funkcji `DwieSumy` z zadania 1 w serii 4, tak aby działał on na wskaźnikach, a nie referencjach.

b) Do funkcji `ord3` z zadania 2b w serii 3 dopisz drugą o tej samej nazwie, która wykonuje to samo, ale przez mechanizm wskaźników, a nie referencji. W funkcji `main` zakoduj sprawdzanie obu przypadków na przykładowych trzech liczbach.

Zadanie 6 (char, wskaźniki i referencje)

Niech użytkownik wpisze z klawiatury znak do zmiennej `char c`.

(a) Napisz funkcję `void Zamien`, która przejmuje zmienną `char` metodą referencji. Jeśli otrzymała literę małą, niech ją zamieni na wielką, a jeśli otrzymała wielką – to na małą. W funkcji `main` wywołaj `Zamien`, a następnie wypisz wynik na ekran.

(b) Przeciąż powyższą funkcję tak, aby wykonywała to samo, jednak przyjmowała wskaźnik na zmienną typu `char`. W funkcji `main` przekopiuj oryginalny znak z klawiatury do nowej zmiennej. Na tej kopii zastosuj nowy wariant `Zamien`, a następnie wypisuj również ten wynik na ekran.

Zadanie 7 (wskaźnik funkcyjny)

(a) Napisz funkcję `Bisection` pracującą na typie `double`, której zadaniem będzie znalezienie miejsca zerowego funkcji $f(x) = exp(-x) - x$ w przedziale `[x1, x2]` = [0 , 1] z dokładnością do `eps = 1e-10`. Niech `x1, x2` i `eps` będą podawane w argumentach wejścia funkcji `Bisection`.

Funkcję $f(x)$ zaimplementuj w osobnej funkcji: `double myfun (double x)`.

Następnie zmień `Bisection` tak, aby pracowała na wskaźniku funkcyjnym typu `double (*myfunptr) (double)`. W `main` zmodyfikuj wywołanie tak, aby podać wskaźnik dostępnej funkcji `myfun`.

Możesz użyć biblioteki `<climits>` albo `<limits>` do ustawienia epsilona i zakresu wyświetlanych cyfr. W tej drugiej bibliotece mamy:

```
cout << setprecision ( numeric_limits<double>::max_digits10 )
     << Bisection (myfun, 0.,1. , numeric_limits<double>::epsilon() )
     << endl;
```

Zadanie 8 (wskaźnik funkcyjny)

Przekształć zadanie 4 z serii 2 tak, aby funkcja `Integral` pracowała na wskaźniku funkcyjnym, który będzie przyjmować przez dodatkowy argument wejścia. W ten sposób stanie się ona ogólna. W funkcji `main` wywołaj `Integral`, najpierw podając na wejście funkcję `fun`, a następnie wywołaj, podając nazwę jakiejś funkcji analitycznej z `cmath` (np. `cos`, `log`, ...)

Zadanie 9 (wskaźnik funkcyjny)

Przekształć kod z zad. 2, tak aby funkcja `Analiza` przyjmowała wskaźnik na dowolną funkcję przyjmującą `double` i zwracającą `double` (np. o nazwie `funptr`). W funkcji `main` wywołaj funkcję `Analiza` dwa razy: za pierwszym razem podając Twoją funkcję matematyczną w kodzie, a za drugim – jedną z funkcji analitycznych z biblioteki `cmath`.