Programowanie Zaawansowane FM i NI
Ćwiczenia 5

Zadanie 1 (Tablice statyczne, algorytmika)

Napisz funkcję `eratostenes`, znajdującą wszystkie liczby pierwsze mniejsze od danej liczby naturalnej `Nmax` – metodą sita Eratostenesa. Program powinien czytać tę liczbę z klawiatury i wypisywać wynik.

Do algorytmu wykorzystaj tablice statyczne. Załóż, że maksymalnym wymiarem tablicy jest 5000. Na dalszym etapie zajęć zapoznasz się z tablicami o wymiarze elastycznym („alokacja dynamiczna").

Nb. najprostszym poleceniem inicjującym tablicę do 0 jest: `typ nazwa[wymiar] = {};`

Zadanie 2 (Statyczne tablice 2-wymiarowe, algorytmika)

Napisz kod wyznaczający wyrazy trójkąta Pascala do jego 12. wiersza. Do przechowania danych użyj 2-wymiarowych tablic statycznych. Następnie wypisz wyrazy trójkąta na ekran.

Zadanie 3 (Przekazywanie tablicy przez wskaźnik)

Napisz funkcję o prototypie:

```
const double* aver (const double* arr, size_t size, double& average);
```

która pobiera tablicę liczb typu double i jej wymiar oraz zmienną average typu double przez referencję. Zadaniem tej funkcji jest:
– wstawienie do zmiennej `average` średniej arytmetycznej elementów tablicy
– zwrócenie adresu tego elementu tablicy, którego wartość jest najbliższa tej średniej.

Na przykład następujący program:

```
#include <iostream>
#include <cmath>
using namespace std;

const double* aver (const double* arr, size_t size, double& average) {
  // ...
}

int main () {
  double arr[] = { 1, 7, 5, 4, 3, 2 };
  size_t size = sizeof(arr) / sizeof(arr[0]);
  double average = 0;
  const double* p = aver (arr, size, average);
  cout << *p << " " << average << endl;
}
```

powinien wypisać:

```
 4 3.66667
```

Zadanie 4 (Porządkowanie tablicy imion)

Na podstawie tego wzorca, napisz funkcję, który posortuje tablicę imion (C-stringów) pod kątem wzrastającej liczby znaków w imieniu. Proponowane kroki pośrednie:

1) Odśwież sortowanie bąbelkowe na przykładzie jakiejś tablicy liczb.
2) Napisz funkcję `void Wypisz (char* Imiona[], int size)`, wypisującą na ekran imiona z tablicy.
3) W funkcji `main` dostępna jest tablica imion jako tablica adresów kolejnych napisów. Napisz taki wariant sortowania, aby dla pary napisów porównywane były ich długości, a zamieniane – ich adresy w tablicy imion. Wypisz na ekran imiona przed i po sortowaniu.
4) Przenieś sam algorytm sortowania do funkcji `void Sortuj (char* Imiona[], int size)`. Ponieważ tablica przenosi się poprzez adres, to w `Sortuj` dysponujesz oryginalną tablicą (możesz to sprawdzić, porównując adresy).

Uwaga: Nie stosuj żadnych innych typów obiektów, kolekcji ani referencji.

Wskazówka: liczbę znaków w napisie zwraca funkcja `strlen` z biblioteki `cstring` . Możesz użyć funkcję `swap`.

Zadanie 5 (argumenty wywołania programu)

W tym wzorcu zakodowany jest kalkulator dwuelementowy, który prosi użytkownika o wskazanie działania poprzez podanie symbolu, a następnie prosi o 2 liczby. Symbole działań to: { `+ - * / f p h` } , gdzie: `f` = reszta z dzielenia (`fmod`), `p` = potęga (`pow`), `h` = pierwiastek z sumy kwadratów (`hypot`) .

Zmień ten kod tak, aby użytkownik mógł podawać w opcjach wywołania kodu dane, np. tak:

```
./kalkulator 1.23 / -4.56
```

Sprawdzaj na początku, czy liczba opcji w wywołaniu jest prawidłowa: jeśli nie, to poinformuj użytkownika o sposobie wywołania i zakończ program.

Uwaga: chcąc przemnożyć liczby, należy wpisać: `\*` . Przyczyną jest konieczność oprotegowania znaku `*` , tak aby system operacyjny nie uznał go za znak kontrolny.

Zadanie 6 (string – wyszukiwanie)

Korzystając z poleceń wyszukiwania w stringu, policz liczbę zdań i linii w tym napisie:

```
string s = "- Programowanie jest proste - stwierdzila Monika.\n"
           "- Serio? - zapytal Rafal.\n"
           "- Serio - odpowiedziala.\n";
```

Przyjmij, że zdaniem jest każdy zbiór znaków kończący się kropką lub znakiem zapytania, a każda linia kończy się znakiem kontrolnym `\n` (nb. w tablicy ASCII ten znak ma wartość 10, ale nie musisz tego faktu używać).

Następnie zamień `Monika` na `Agnieszka`, a `Rafal` na `Patryk`, nie psując zdania.

Uwaga: Nie przechodź stringu znak po znaku, pozwól dedykowanej funkcji to wykonać.

(b) zmodyfikuj kod tak, aby imiona, na które ma się dokonać zamiana, wpisywał użytkownik w argumentach wywołania kodu. Sprawdzaj, czy liczba tych argumentów jest właściwa. Jeśli nie, to wyświetl poradę, jak używać kodu i przerwij program.