TERMINY ZAJĘĆ: 9, 16, 23, 30 MARCA
I 6, 13 KWIETNIA 2019 ROKU,
GODZ. 9.00 – 12.15
MIEJSCE: WYDZIAŁ MATEMATYKI I
NAUK INFORMACYJNYCH
SALA: 311

PROWADZĄCY: BARTŁOMIEJ JOP
KONTAKT:
BARTEKJ_97@HOTMAIL.COM

C++
POZIOM PODSTAWOWY
Z ELEMENTAMI ARDUINO

Język programowania: **C/C++**

Środowisko programowania: **Visual Studio**

Arduino.ide

TABLICE DWUWYMIAROWE

- Definiowanie tablicy:

  typ_elementów Nazwa[ilość Wierszy] [ilość Kolumn]

LICZBY LOSOWE

- Instrukcja zwracająca pseudolosową liczbę z przedziału 0-RAND_MAX
- Instrukcja zwraca zawsze ten sam ciąg liczb pseudolosowych
- Zawarta w bibliotece <cstdlib>
- Postać instrukcji: rand()
- Instrukcja srand(time(0)) inicjuje ciąg liczb wartością losową, pobieraną z zegara systemowego, co pozwala otrzymać inne liczby losowe przy każdym uruchomieniu programu (instrukcja time(0) zawarta jest w bibliotece <ctime>)

ZADANIE

Napisz funkcję, która uzupełni tablicę 15 elementową losowymi liczbami z przedziału podanego przez użytkownika.

ZADANIE

Napisz program, który uzupełni tablicę 15 x 5 elementową losowymi liczbami z przedziału podanego przez użytkownika, a następnie wydrukuję ją.

Uwaga: jako odstępy pomiędzy elementami w wierszach zastosuj tabulator

ALGORYTMY PROSTEGO SORTOWANIA

1. Sortowanie przez wybieranie
2. Sortowanie przez wstawianie
3. Sortowanie przez zamianę

SORTOWANIE PRZEZ WYBIERANIE

Zasada działania: Wybierz najmniejszy element z nieposortowanych następnie zamień z pierwszym nieposortowanym – aż do posortowania całości.

SORTOWANIE PRZEZ WSTAWIANIE

Zasada działania: Wybierz pierwszy element z nieposortowanych i wstaw go w odpowiednie miejsce w części posortowanej – aż do posortowania całości.

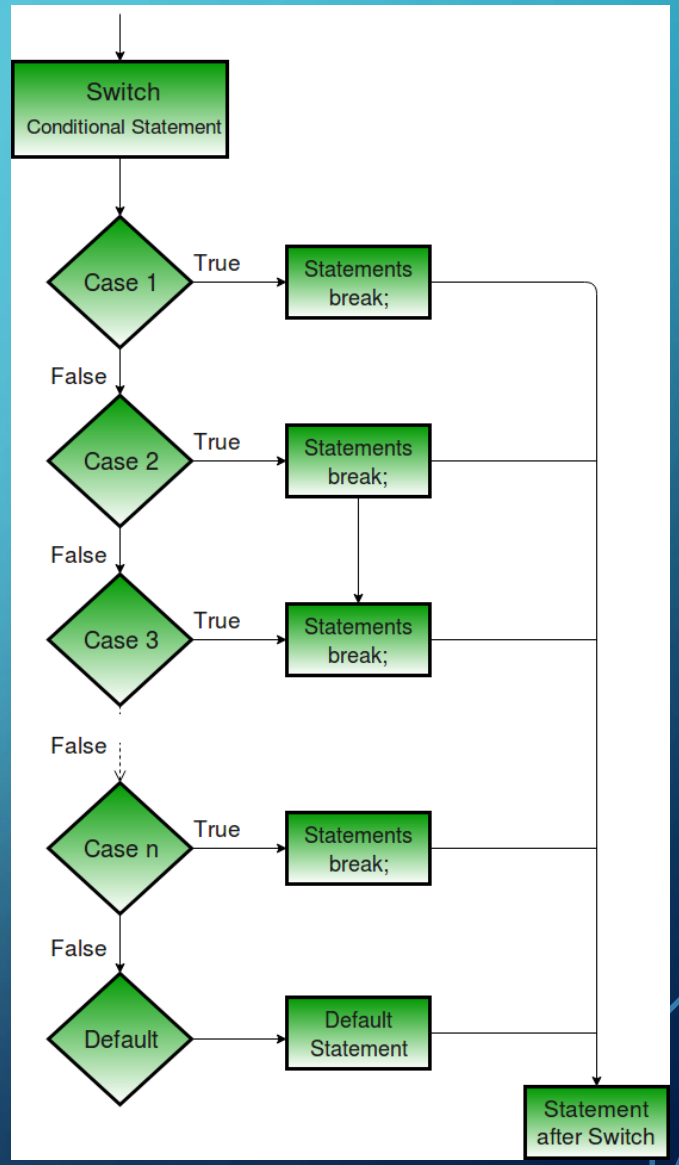
SORTOWANIE PRZEZ ZAMIANĘ

Zasada działania: Zamień ze sobą, jeśli trzeba, dwóch sąsiadów – najpierw w całej tablicy, a potem za każdym razem w części mniejszej o 1 element.

INSTRUKCJA SWITCH

- Jest to wielowymiarowy „if”
- Postać instrukcji:

```
switch(selektor){
        case stała:
                ciąg instrukcji;
                break;
        case stała:
                ciąg instrukcji;
                break;
        case stała:
                ciąg instrukcji;
                break;
        deafult:
                ciąg instrukcji;
```



- Uwaga: instrukcja break przerywa dalsze działanie switcha, nie jest ona konieczna, wtedy instrukcja przejdzie do sprawdzania kolejnych przypadków
- Selektor może mieć typ: int, char, bool(nie może być typu double/string!)

REKORDY

- Umożliwiają przechowywanie wielu różnych zmiennych pod jedną nazwą
- Wymagają zdefiniowania własnej struktury
- Definiowanie struktury:

```
struct Nazwa_struktury{
        typ_zmiennej nazwa_zmiennej;
        typ_zmiennej nazwa_zmiennej;
        …
};
```

Uwaga! Średnik musi się pojawić na końcu definiowania struktury

- Tworzenie rekordu ze struktury:

```
Nazwa_Struktury Rekord;
```

- Po utworzeniu rekordu możemy wczytywać dane do i z danego pola odwołując się do pola w następujący sposób:

```
Rekord.nazwa_zmiennej = wartość;
```

- Struktury można zagnieżdżać w sobie

PLIKI TEKSTOWE

- W takich plikach zawarte są ciągi znaków widocznych oraz znaki niewidoczne
- Ciągi znaków podzielone są na wiersze
- Instrukcje umożliwiające operacje na plikach tekstowych zawarte są w bibliotece <fstream>
- Dostęp do plików otrzymujemy po stworzeniu tzw. zmiennych plikowych
- ifstream plik_odczyt; -utworzenie zmiennej plikowej umożliwiającej odczyt
- ofstream plik_zapis; -utworzenie zmiennej plikowej umożliwiającej zapis

PLIKI TEKSTOWE CD

- Zmienna_plikowa. open (nazwa_pliku) - otwarcie pliku do odczytu lub zapisu
- Zmienna_plikowa. close ( ) - zamknięcie pliku
- Zmienna_plikowa. clear ( ) - reset pliku (przed ponownym otwarciem)
- Zmienna_plikowa. eof ( ) - funkcja typu boolean, która zwraca informację, czy napotkano koniec pliku
- zmienna plikowa. is_open ( )
- zmienna plikowa. good ( ) - zwracają informację typu boolean, czy plik istnieje (i jest otwarty)
- nazwa_pliku – ciąg znaków określający nazwę fizycznego pliku
- przykłady nazw :
  - plik.txt – plik znajdujący się w folderze projektu o nazwie plik i rozszerzeniu .txt
  - "C:/plik1.txt" – plik z podaniem ścieżki
  - "C:\\wyniki.txt" – plik z podaniem ścieżki(uwaga! Dwa znaki \ ponieważ jeden oznacza znak specjalny)
  - Jeśli nazwa zapisana jest w zmiennej to aby otworzyć taki plik należy przy podawaniu nazwy zmiennej dodać .c_str()

WŁASNE FUNKCJE

- Praktycznie każdy fragment kodu można zapisać jako własną funkcję pod jakąś nazwą
- Funkcja wykonując ciąg instrukcji na koniec wyznacza jedną wartość i zwraca ją poprzez nazwę funkcji. Typ zwracanej wartości będziemy nazywali w skrócie typem funkcji.
- Aby funkcja zwróciła jakąś wartość, potrzebna jest instrukcja return, która przerywa natychmiast działanie funkcji przekazując wynik do miejsca wywołania funkcji.
- Definiowanie funkcji:

```
typ_funkcji nazwa_funkcji (parametry wejściowe){
        ciąg instrukcji
        return wartosc
}
```

WŁASNE FUNKCJE CD

- Typ wyniku funkcji może być jednym z następujących:
  - typem prostym (int, double i inne typy liczbowe, char, bool)
  - typem napisowym (string)
  - typem wskaźnikowym
  - typem rekordowym
  - typem pustym (void), który oznacza, że funkcja nie zwraca wyniku, więc jej typ nie jest określony. Taka funkcja służy więc tylko do wykonania ciągu instrukcji.
- Funkcje piszemy przed int main(){}
- Jeśli chcemy jednak umieścić po „mainie” to przed nim musimy umieścić nagłówek funkcji

ZADANIE

Napisz funkcję, która przesunie wszystkie elementy tablicy 5 elementowej w prawo.

ZADANIE

Napisz funkcję, która przesunie wszystkie elementy tablicy 5 elementowej w lewo.

ZADANIE

Napisz funkcję, która przesunie wszystkie elementy tablicy 15 elementowej w lewo o 3 miejsca.

ZADANIE

Napisz program, która obliczy sumę elementów w każdym wierszu, kolumnie, na głównej przekątnej, nad nią oraz pod główną przekątną