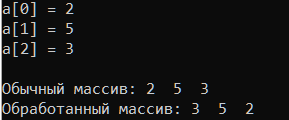
**ЗВІТ ДО ПР №6-7**

*Виконав студент 35 групи*

*Рижій Ілля*

**Завдання 1.** Скласти функцію, яка переставляє місцями два елементи масиву із заданими номерами.



Код програми:

#include <stdio.h>

#include <locale.h>

int main()

{

setlocale(LC\_ALL, "rus");

int i;

int n = 3;

int arr[n];

for(i = 0; i < n; i++){

printf("a[%d] = ", i);

scanf("%d", &arr[i]);

}

printf("\nОбычный массив:");

for(i = 0; i < n; i++){

printf(" %d ", arr[i]);

}

printf("\nОбработанный массив:");

for (i = 0; i < n + 1; i++){

int temp = arr[i];

arr[i] = arr[i + 2];

arr[i + 2] = temp;

for (i = 0; i < n; i++){

printf(" %d ", arr[i]);

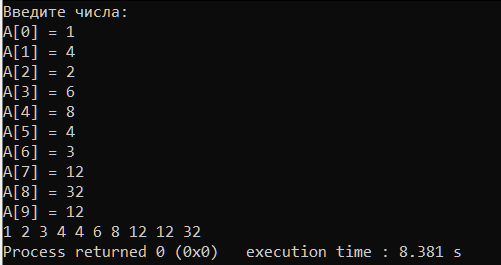
}

}

return 0;

}

**Завдання 2.** Скласти програму Selection.cpp, що забезпечує сортування масиву методом прямого вибору.



Код програми:

#include <stdio.h>

#include <conio.h>

#include <locale.h>

void Sorting(int \*arr, int size);

int main()

{

setlocale(LC\_ALL, "rus");

int N = 10;

int A[N], i;

printf("Введите числа: \n");

for(i = 0; i < N; i++)

{

printf("A[%i] = ", i);

scanf("%i", &A[i]);

}

Sorting(A, N);

return 0;

}

void Sorting(int \*arr, int size) // функція сортування

{

int min, buf; // змінні мінімального пошуку і тимчасового збереження

for (int i = 0; i < size - 1; i++)

{

min = i;

// пошук мінімального елементу

for (int j = i + 1; j < size; j++)

{

if (arr[j] < arr[min]) // якщо елемент менше мінімального,

min = j;

}

buf = arr[i]; // змінюємо місцями і-ий і мінімальний елемент

arr[i] = arr[min];

arr[min] = buf;

}

for(int i = 0; i < size; i++)

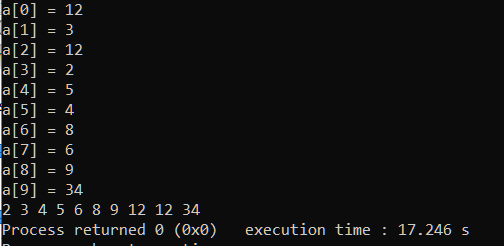
{

printf("%i ", arr[i]);

}

}

**Завдання 3.** Скласти програму BubleSort.cpp, що забезпечує сортування масиву за алгоритмом "бульбашки".



Код програми:

#include <stdio.h>

#include <conio.h>

#include <locale.h>

void Sorting(int \*arr, int size);

int main()

{

setlocale(LC\_ALL, "rus");

int A[10], i;

for (i = 0; i < 10; i++)

{

printf("a[%d] = ", i);

scanf("%d", &A[i]);

}

Sorting(A, 10); // виклик функції сортування

getch();

}

void Sorting(int \*arr, int size)

{

int temp, i, j;

for (i = 0; i < size - 1; i++)

{

for (j = (size - 1); j > i; j--)

{

if (arr[j - 1] > arr[j]) // якщо елемент менший за попередній

{

temp = arr[j - 1]; // міняємо місцями

arr[j - 1] = ar[j];

arr[j] = temp;

}

}

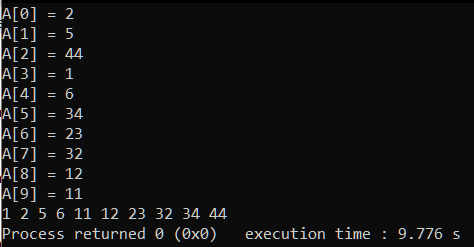
}

for (i = 0; i < 10; i++)

printf("%d ", arr[i]);

}

**Завдання 4.**  Змінити програму, забезпечивши реалізацію шейкерного сортування. Збережіть під назвою ShakerSort.cpp.



Код програми:

#include <stdio.h>

#include <conio.h>

#include <locale.h>

void Sorting(int \*arr, int count);

int main() {

setlocale(LC\_ALL, "rus");

int Arr[10];

for (int i = 0; i < 10; i++) {

printf("A[%i] = ", i);

scanf("%i", &Arr[i]);

}

Sorting(Arr, 10);

return 0;

}

void Sorting(int \*arr, int count)

{

int left = 0, right = count - 1; //

int flag = 1;

// доки ліва граниниця не зустрінеться з правою

// і доки в масиві є переміщення

while ((left < right) && flag > 0)

{

flag = 0;

for (int i = left; i < right; i++) //рух зліва направо

{

if (arr[i] > arr[i + 1]) // якщо наступний елемент менший за поточний, то міняємо їх місцями

{

int t = arr[i];

arr[i] = arr[i + 1];

arr[i + 1] = t;

flag = 1;

}

}

right--; // зрушуємо праву границю на попередній елемент

for (int i = right; i>left; i--) //

{

if (arr[i - 1]>arr[i])

{

int t = arr[i];

arr[i] = arr[i - 1];

arr[i - 1] = t;

flag = 1;

}

}

left++;

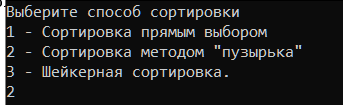
}

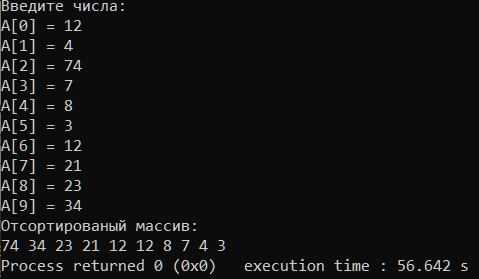
for (int i = 0; i < 10; i++)

printf("%i ", arr[i]);

}

**Завдання 5.** Скласти програму, що забезпечує сортування масиву (у порядку спадання) одним із вибраних методів (оператор вибору). Номер методу вводиться з клавіатури.





Код програми:

#include <stdio.h>

#include <conio.h>

#include <locale.h>

void StraightSorting(int \*arr, int size) // функція сортування

{

int min, buf; // змінні мінімального пошуку і тимчасового збереження

for (int i = 0; i < size - 1; i++)

{

min = i;

// пошук мінімального елементу

for (int j = i + 1; j < size; j++)

{

if (arr[j] > arr[min]) // якщо елемент більший за мінімальний,

min = j;

}

buf = arr[i]; // змінюємо місцями і-ий і мінімальний елемент

arr[i] = arr[min];

arr[min] = buf;

}

printf("Отсортированый массив:\n");

for(int i = 0; i < size; i++)

{

printf("%i ", arr[i]);

}

}

void BubbleSorting(int \*arr, int size)

{

int temp, i, j;

for (i = 0; i < size - 1; i++)

{

for (j = (size - 1); j > i; j--)

{

if (arr[j - 1] < arr[j]) // якщо елемент більший за попередній

{

temp = arr[j - 1]; // міняємо місцями

arr[j - 1] = arr[j];

arr[j] = temp;

}

}

}

printf("Отсортированый массив:\n");

for (i = 0; i < 10; i++)

printf("%d ", arr[i]);

}

void ShakeSorting(int \*arr, int count)

{

int left = 0, right = count - 1; //

int flag = 1;

// доки ліва граниниця не зустрінеться з правою

// і доки в масиві є переміщення

while ((left < right) && flag > 0)

{

flag = 0;

for (int i = left; i < right; i++) //рух зліва направо

{

if (arr[i] < arr[i + 1]) // якщо наступний елемент більший за поточний, то міняємо їх місцями

{

int t = arr[i];

arr[i] = arr[i + 1];

arr[i + 1] = t;

flag = 1;

}

}

right--; // зрушуємо праву границю на попередній елемент

for (int i = right; i > left; i--) //

{

if (arr[i - 1] < arr[i])

{

int t = arr[i];

arr[i] = arr[i - 1];

arr[i - 1] = t;

flag = 1;

}

}

left++;

}

printf("Отсортированый массив:\n");

for (int i = 0; i < 10; i++)

printf("%i ", arr[i]);

}

int main()

{

setlocale(LC\_ALL, "rus");

const int N = 10;

int choice, i, A[N];

printf("Выберите способ сортировки\n"

"1 - Сортировка прямым выбором\n"

"2 - Сортировка методом \"пузырька\"\n"

"3 - Шейкерная сортировка.\n");

scanf("%i", &choice);

system("cls");

switch(choice)

{

case 1:

printf("Введите числа: \n");

for(i = 0; i < N; i++)

{

printf("A[%i] = ", i);

scanf("%i", &A[i]);

}

StraightSorting(A, N);

break;

case 2:

printf("Введите числа: \n");

for(i = 0; i < N; i++)

{

printf("A[%i] = ", i);

scanf("%i", &A[i]);

}

BubbleSorting(A, N);

break;

case 3:

printf("Введите числа: \n");

for(i = 0; i < N; i++)

{

printf("A[%i] = ", i);

scanf("%i", &A[i]);

}

ShakeSorting(A, N);

break;

default:

printf("Такой функции нету((");

}

}

**Висновки:**  під час практичної роботи узагальнили знання про алгоритми сортування масивів.