Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

**Алгоритмізація та програмування 1: Базові концепції програмування**

ЗВІТ

до лабораторної роботи №7+8

«**алгоритми перестановок елементів масивів та способи сортування масиві**»

*(ТЕМА)*

Варіант № 1

Дата «10» грудня 2021 Виконав: студент 1 курсу

гр. ТР-15

Руденко Владислав Ігорович

*(П.І.Б.)*

Оцінка «\_\_\_\_\_\_\_\_\_» Перевірив:

Крячок Олександр Степанович

Дата «\_\_\_» \_\_\_\_\_\_\_ 20\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(П.І.Б., підпис)*

Київ – 2021

***Завдання до роботи:***

1. Ознайомитись з алгоритмами перестановок елементів масивів та способами сортування масивів.  
2. У якості індивідуального завдання необхідно написати програмний код, що реалізує перестановку елементів масивів та сортування масивів.  
3. Звернення до елементів масиву реалізувати за допомогою вказівника на масив.  
4. Роздрукувати (вивести на екран) початковий масив та масив після виконання сортування.

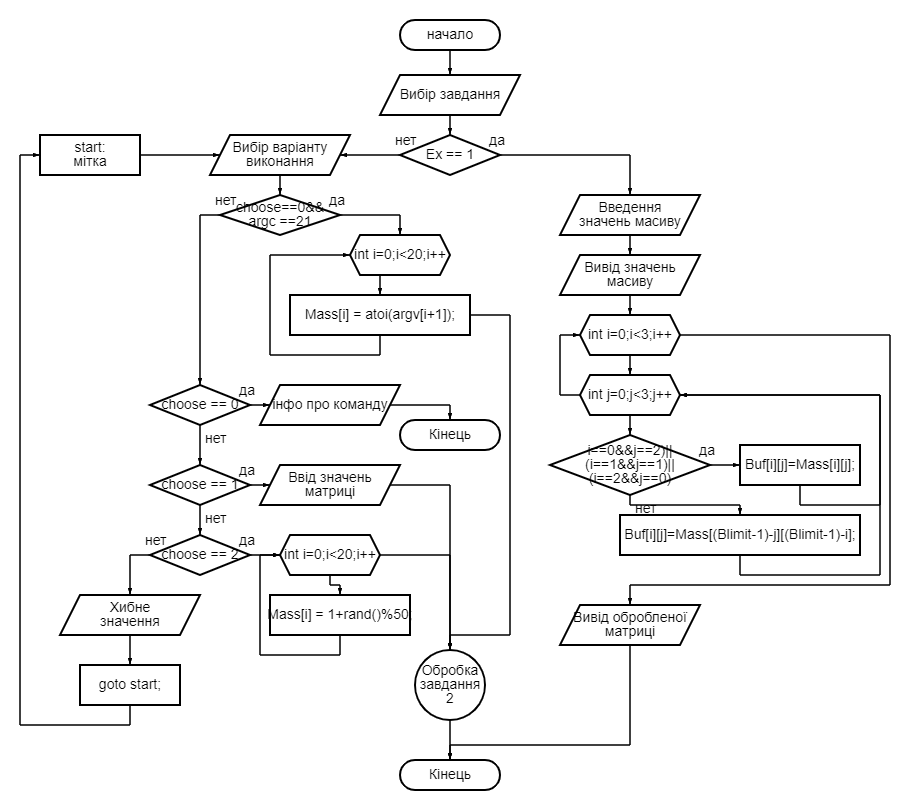
***Теоритичні відомості:***

**Сортування бульбашкою.**  
Алгоритм складається з повторюваних проходів по масиву, що сортується. За кожен прохід елементи послідовно порівнюються попарно і якщо порядок у парі неправильний, виконується перестановка елементів. Проходи по масиву повторюються N-1 разів або до тих пір, поки на черговому проході не виявиться, що обміни більше не потрібні, що означає – масив відсортований. При кожному проході алгоритму за внутрішнім циклом, черговий найбільший елемент масиву ставиться на своє місце в кінці масиву поруч із попереднім «найбільшим елементом», а найменший елемент переміщується на одну позицію до початку масиву («спливає» до потрібної позиції, як пляшечка у воді. звідси і назва алгоритму.

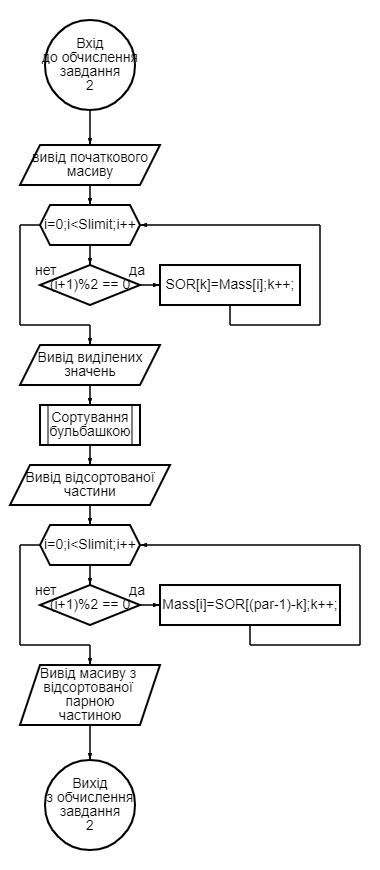
Особливість даного алгоритму полягає в наступному: після першого завершення внутрішнього циклу максимальний елемент масиву завжди знаходиться на N-ої позиції. При другому проході максимальний елемент знаходиться на N-1 місці. І так далі. Таким чином, на кожному наступному проході кількість оброблюваних елементів зменшується на 1 і немає необхідності обходити весь масив від початку до кінця кожен раз.

**Вказівники**

У Cі вказівники та масиви тісно пов'язані. Зазвичай компілятор перетворює масив навказівники. За допомогою вказівники можна маніпулювати елементами масиву, як і за допомогою індексів. Вказівниками складно керувати. Досить легко записати в вказівник неправильне значення, що може викликати помилку, що важко відтворюється. Наприклад, ви випадково змінили адресу вказівника в пам'яті, або неправильно виділили під інформацію пам'ять і тут на вас може очікувати сюрприз: інша дуже важлива змінна, яка використовується тільки всередині програми буде перезаписана. Зрозуміти, де саме є помилка і відтворити її буде нелегко, а усунення таких помилок — який завжди тривіальна завдання, іноді доводиться переписувати істотну частину программы

***Блок схема алгоритму:  
***

Блок-схема 1 Основна частина



Блок-схема 2 Обробка для завдання 2

***Результати роботи:***

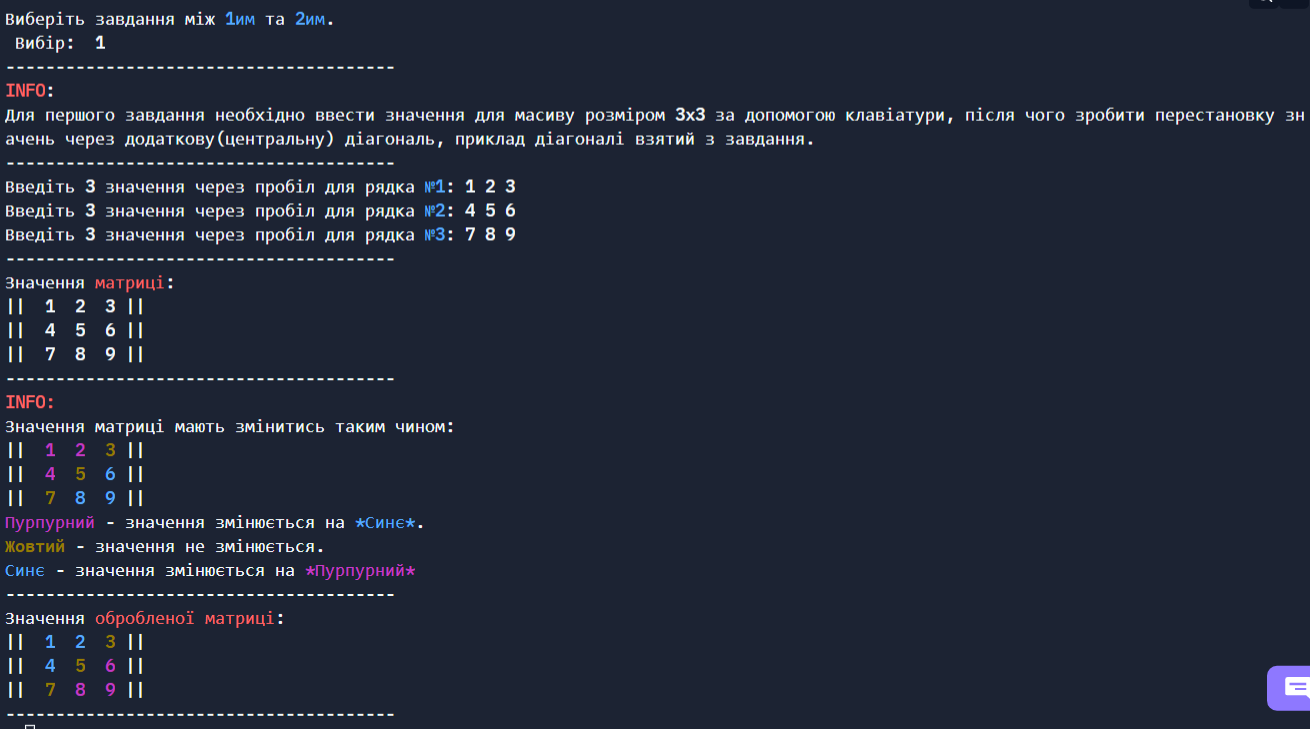


Рисунок 1 Завдання 1 при виведені додаткової інформації

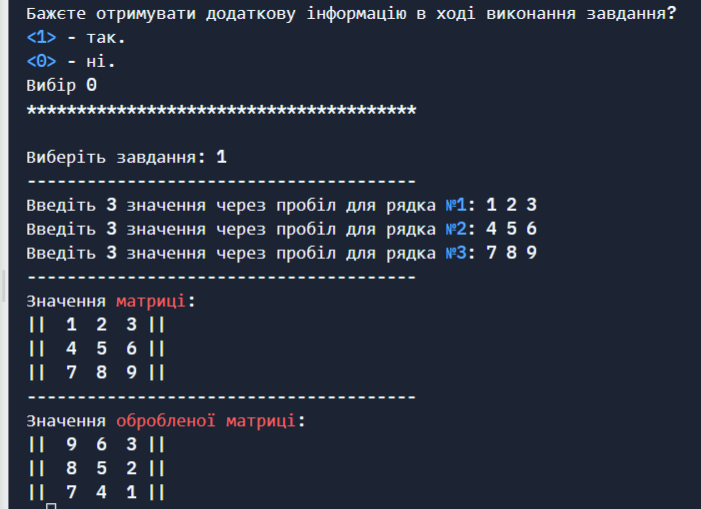


Рисунок 2 Завдання 1 при звичайному виводі

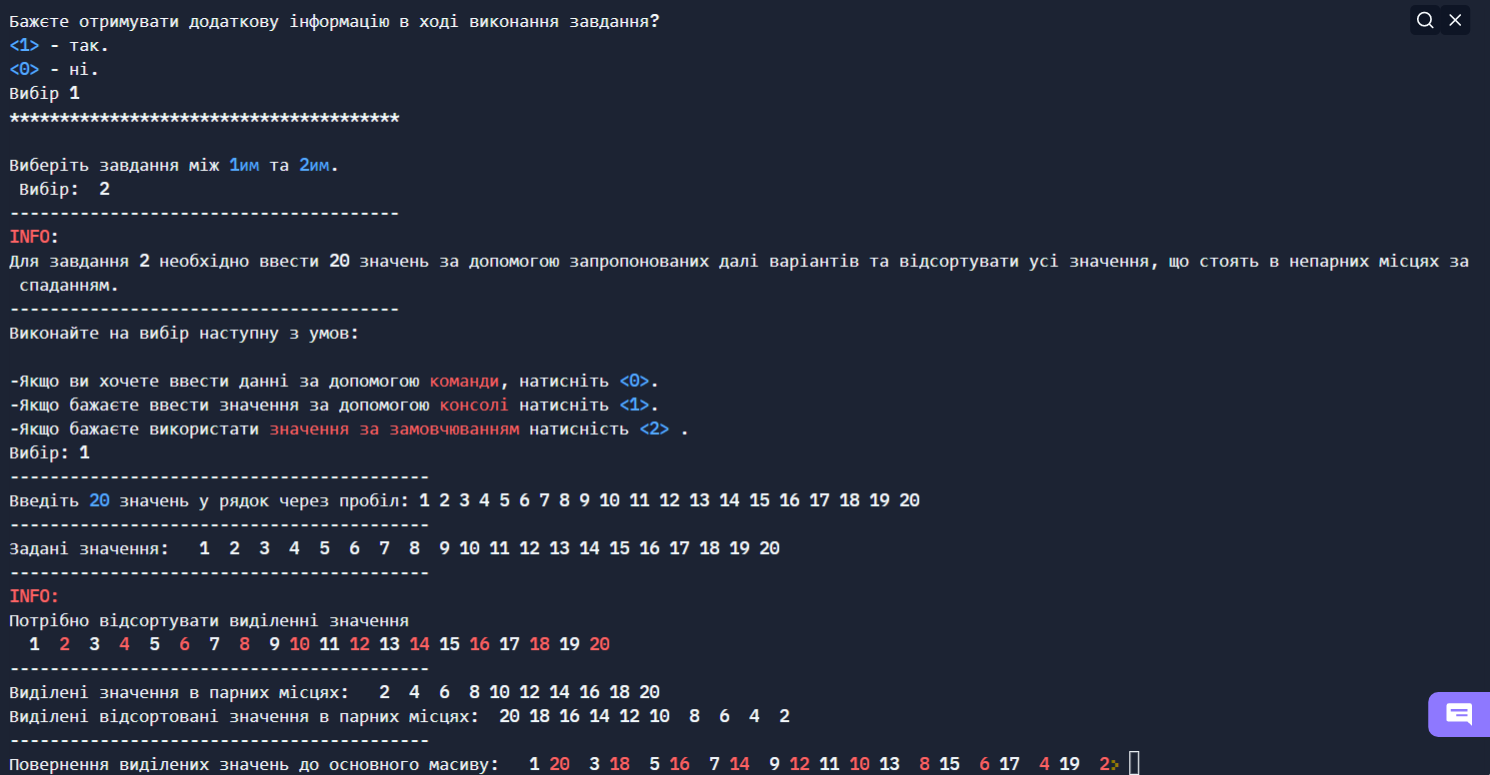


Рисунок 3 Завдання 2 при виведені додаткової інформації

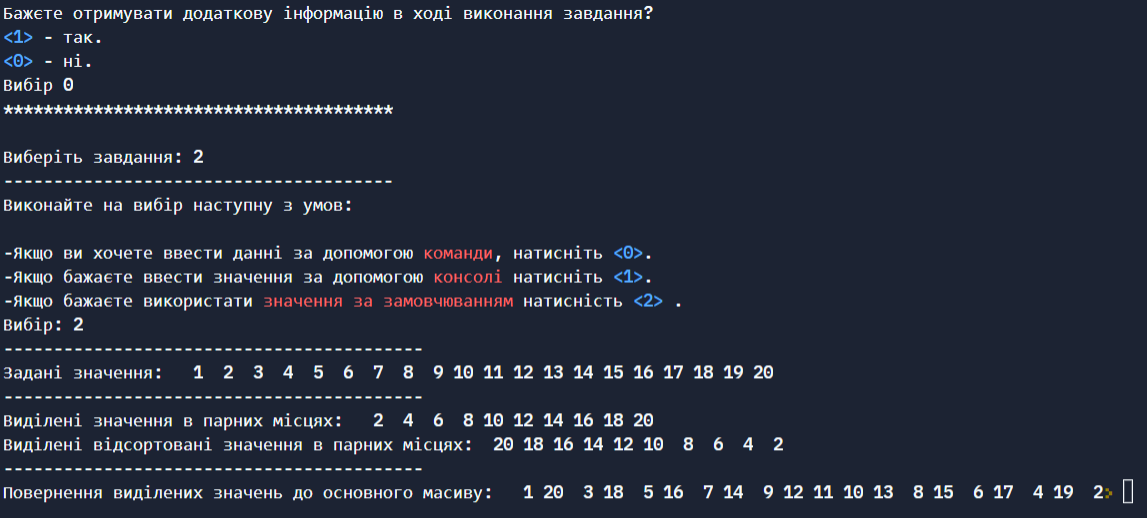


Рисунок 4 Завдання 2 при звичайному виводі

***Хід виконаної роботи:***

Спочатку користувачу дається вибір на виконання алгоритму з виведенням додаткової інформації та без(Рекомендуєтсья виконувати з виведенням додаткової інформації) після чого йде вибір завдання, для завдання 1 необхідно ввести 9 значень масиву (по 3 в кожен рядок) після чого програма перетворює масив з переміщенням через додаткову діагональ, результати виводяться на єкран. Для завдання 2 користувачу надається 3 варіанти введення (оскільки умовою не передбачено конкретний тип) після чого заданий масив з 20 чисел виводиться користувачу, з парних місць масиву беруться дані та переносяться в буферний масив, цей масив сортується за спаданням та виводиться користувачу, додатково значення буферного масиву вводяться в оригінальний масив та виводяться користувачи у вигляді суцільного масиву з відсортованими елементами.

***Посилання на програму в repl.it:***

***<https://replit.com/join/dfrbeochsm-hetik>***

***Висновки:***

В ході виконання лабораторної роботи № 7+8 було ознайомленно з  алгоритмами перестановок елементів масивів та способами сортування масивів.  У якості індивідуального завдання було написати програмний код, що реалізує перестановку елементів масивів та сортування масивів. В ході виконання було використано декілька методів реалізації початкових данних, також було виявленно та виправлено декілька помилок

***Програмний код (у текстовому варіанті):***

//LRC\_7+8//Руденко Владислав//ТР-15//9.12//

///////////////////////////////////////////

#include <stdio.h>

#include <math.h>

#include <stdlib.h>

#define RED "\x1b[31m"

#define BLUE "\x1b[34m"

#define PUR "\x1b[35m"

#define YELLOW "\033[1;33m"

#define RESET "\x1b[0m"

#define Blimit 3

#define Slimit 20

#define par 10

int main(int argc, char \*argv[])

{

int info = 1;

if(argc == 21)

{goto flash;}

int Ex;

//Чи бажає користувач мати розширений вигляд коду

printf(RED"System message"RESET":\nБажєте отримувати додаткову інформацію в ході виконання завдання?\n"BLUE"<1>"RESET" - так. \n"BLUE"<0>"RESET" - ні.\nВибір ");

scanf("%i", &info);

if(info>1 || info <0)

{

printf("Відповідь сприйнята як "BLUE"<0>"RESET"\n");

info = 0;

}

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");

start:

//ВИбір завдання

if(info == 0) printf("Виберіть завдання: ");

else if(info == 1) printf("Виберіть завдання між "BLUE"1им"RESET" та "BLUE"2им"RESET".\n Вибір: ");

scanf("%i",&Ex);

printf("---------------------------------------\n");

if(Ex == 1)

{//Початок виконання завдання №1

int Mass[Blimit][Blimit];

int Buf[Blimit][Blimit];

if(info == 1)

{

printf(RED"INFO"RESET":\nДля першого завдання необхідно ввести значення для масиву розміром %ix%i за допомогою клавіатури, після чого зробити перестановку значень через додаткову(центральну) діагональ, приклад діагоналі взятий з завдання.\n---------------------------------------\n", Blimit, Blimit);

}

for(int i = 0;i < 3;i++)

{//Ввід вхідних значень

printf("Введіть 3 значення через пробіл для рядка "BLUE"№%d"RESET": ",i+1);

for(int j = 0;j < 3;j++)

{

scanf("%i", &Mass[i][j]);

}

printf("");

}//БАзовий вивід матриці

printf("---------------------------------------\nЗначення "RED"матриці"RESET": \n");

for(int i =0; i<3; i++)

{

printf("||");

for(int j=0;j<3;j++)

{

printf("%3i", Mass[i][j]);

}

printf(" ||\n");

}

if(info ==1)

{//якщо користувач вибрав розширене інфо

printf("---------------------------------------\n");

printf(""RED"INFO:"RESET"\nЗначення матриці мають змінитись таким чином: \n");

for(int i =0; i<3; i++)

{

printf("||");

for(int j=0;j<3;j++)

{

if((i==0&&j<2)||(i==1&&j==0)) printf(PUR);

else if((i==0&&j==2)||(i==1&&j==1)||(i==2&&j==0)) printf(YELLOW);

else if((i==2&&(j<3&&j>0))||(i==1&&j==2))printf(BLUE);

printf("%3i", Mass[i][j]);

printf(RESET);

}

printf(" ||\n");

}

printf(""PUR"Пурпурний"RESET" - значення змінюється на "BLUE"\*Синє\*"RESET".\n"YELLOW"Жовтий"RESET" - значення не змінюється.\n"BLUE"Синє"RESET" - значення змінюється на "PUR"\*Пурпурний\*"RESET"\n");

}

printf("---------------------------------------\n");

//ВИконання поставленої задачі, а саме перестановка значень

for(int i=0;i<3;i++)

for(int j=0;j<3;j++)

{

if((i==0&&j==2)||(i==1&&j==1)||(i==2&&j==0))

Buf[i][j]=Mass[i][j];

else

Buf[i][j]=Mass[(Blimit-1)-j][(Blimit-1)-i];

}

if(info ==1)

{

//Вивід з розширеним інфо

printf("Значення "RED"обробленої матриці"RESET": \n");

for(int i =0; i<3; i++)

{

printf("||");

for(int j=0;j<3;j++)

{

if((i==0&&j<2)||(i==1&&j==0)) printf(BLUE);

else if((i==0&&j==2)||(i==1&&j==1)||(i==2&&j==0)) printf(YELLOW);

else if((i==2&&(j<3&&j>0))||(i==1&&j==2))printf(PUR);

printf("%3i", Mass[i][j]);

printf(RESET);

}

printf(" ||\n");

}

printf("---------------------------------------\n");

}

else

{//Звичайний вивід

printf("Значення "RED"обробленої матриці"RESET": \n");

for(int i =0; i<3; i++)

{

printf("||");

for(int j=0;j<3;j++)

{

printf("%3i",Buf[i][j]);

}

printf(" ||\n");

}

}

}

else if(Ex == 2)

{

int i,choose, temp;

if(info==1)//Розширене інфо

printf(""RED"INFO"RESET":\nДля завдання 2 необхідно ввести 20 значень за допомогою запропонованих далі варіантів та відсортувати усі значення, що стоять в непарних місцях за спаданням.\n---------------------------------------\n");

//Значення за замовчюванням

int Array[Slimit]={1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20};

int \*Mass=Array;//Вибір виконання

printf("Виконайте на вибір наступну з умов:\n\n-Якщо ви хочете ввести данні за допомогою "RED"команди"RESET", натисніть"BLUE" <0>"RESET".\n-Якщо бажаєте ввести значення за допомогою "RED"консолі"RESET" натисніть "BLUE"<1>"RESET".\n-Якщо бажаєте використати "RED"значення за замовчюванням"RESET" натисність "BLUE"<2>"RESET" .\nВибір: ");

scanf("%d", &choose);

printf("------------------------------------------\n");

flash:

if(choose==0&&argc ==21)

{//Присвоєння командою

for(int i=0;i<20;i++)

Mass[i] = atoi(argv[i+1]);

}

else if(choose == 0) {//Завжди виконуєть додаткове інфо\*\* строка 19

if(info == 0)//Інфо стосовно команди

printf("Введіть команду:\n");

else

printf("Введіть команду(./main (20 значень масиву через пробіл)):\n");

return 0;

}

else if(choose == 1)

{//Присвоєння клавіатурою

printf("Введіть "BLUE"20"RESET" значень у рядок через пробіл: ");

for(int i=0;i<20;i++)

scanf("%3d", &Mass[i]);

printf("------------------------------------------\n");

}

else if(choose == 2);

else if(choose>2||choose<0)

{

error:

printf("Error. Неправильне значення");

goto start;

}//Вивід значеннь

printf("Задані значення: ");

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

printf("%3i",Mass[i]);

printf("\n------------------------------------------\n");

if(info == 1)

{//Доп.інфо

printf(RED"INFO:"RESET"\nПотрібно відсортувати виділенні значення\n");

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

{

if((i+1) %2 == 0) printf(RED);

printf("%3i"RESET"",Mass[i]);

}

printf("\n------------------------------------------\n");

}

//Відокремлення парних частин

int SOR[Slimit],k;

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

if((i+1)%2 == 0)

{SOR[k]=Mass[i];k++;}

printf("Виділені значення в парних місцях: ");

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

printf("%3i",SOR[i]);

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

{

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

if (SOR[j] > SOR[j + 1])

{

temp = SOR[j];

SOR[j] = SOR[j+1];

SOR[j+1] = temp;

}

}

printf("\nВиділені відсортовані значення в парних місцях: ");

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

printf("%3i",SOR[(par-1)-i]);

printf("\n------------------------------------------\n");

k=0;

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

if((i+1)%2 == 0)

{Mass[i]=SOR[(par-1)-k];k++;}

printf("Повернення виділених значень до основного масиву: ");

if(info == 1)

{//Доп Інфо

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

{

if((i+1) %2 == 0) printf(RED);

printf("%3i"RESET"",Mass[i]);

}

}

else

{//Звичайни виаід

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

printf("%3i",Mass[i]);

}

}

else if(Ex == 15)

{

//Зайчик

printf(""RED"---------------------------------------\n---------------------------------------\n (\\\_/)\n (>.<)\t\tZai4uk xo4e maximum baliv\n(\")\_(\")\n---------------------------------------\n---------------------------------------\n"RESET);

goto start;

}

else

{

printf("Error... Вибране хибне значення, зробіть вибір ще раз!\n");

goto start;

}

}