

**Звіт з лабораторної роботи
за дисципліною "алгоритми і структури
даних"
студента групи ПА-17-1
Панасенка Егора Сергійовича
Кафедра комп'ютерних технологій, фпм, дну
2017/2018 навч.р.**

1. Постановка задачі:

1. Розробити програму, в якій створюються динамічні масиви, та виконати їх обробку у відповідності до свого варіанту.
 2. Програма повинна містити такі функції.
 1. Введення розміру масиву та формування одновимірного масиву за допомогою команди new.
 2. Заповнення масиву за допомогою датчика випадкових чисел. Елементами масиву мають бути цілі числа у діапазоні, границі якого ввести заздалегідь.
 3. Виведення масиву на екран.
 4. Дії згідно з завданням варіанту. Виведення результатів на екран.
 5. Видалення динамічних масивів за допомогою операції delete.
 3. Організувати роботу програми таким чином, щоб перелічені вище дії можна було б виконувати неодноразово за один сеанс роботи програми.
 4. Варіант 17
 1. Завдання 1. Сформувати новий масив з додатних парних елементів даного масиву, що розташовані після максимального елемента масиву.
 2. Завдання 2. Видалити всі елементи з непарними індексами.
2. Опис структури програми та реалізованих функцій:
1. Програма має такі функції:
 1. void ign_other(FILE * input)
 1. Ігнорує непотрібні данні до кінця рядка. Наприклад якщо ми запросили одне число, а ввели число і якийсь текст то програма забере тільки число.
 2. Аргументи:

1. input - вхідний потік.
3. Функція нічого не виводить.
2. char menu (FILE * input, FILE * output, char * message, char count)
 1. Циклічно запитує меню доки не отримає правильну відповідь.
 2. Аргументи
 1. input – вхідний потік
 2. output – вихідний потік
 3. message – повідомлення яке показується перед вибором, у якому перелічені пункти
 4. count – максимальне число яке можна ввести.
 3. Функція виводить відповідь у вигляді числа від -127 до 128.
3. int * create_array(int length)
 1. Створює динамічний масив розміру length та виводить його адресу.
 2. Аргументи
 1. length – розмір масиву.
 3. Функція виводить адресу
4. void fill_array(int * array, int length, int min, int max)
 1. Заповнює масив array розміру length випадковими числами від min до max.
 2. Аргументи:
 1. array – адреса масиву.
 2. length – розмір масиву.
 3. min – мінімально можливе число.
 4. max – максимально можливе число.
 3. Функція нічого не виводить.
5. void show_array(FILE * output, int * array, int length)
 1. Виводить масив array розміру length у потік output
 2. Аргументи:
 1. output – вихідний потік.
 2. array – адреса масиву.

3. length – розмір масиву.

3. Функція нічого не виводить.

6. int find_max(int * array, int length)

1. Знаходить максимальний елемент у масиві array розміру length

2. Аргументи:

1. array – адреса масиву.

2. length – розмір масиву.

3. Функція виводить позицію максимального елемента.

7. int * task1(int * array, int * length, int max_pos)

1. Сформовує новий масив з додатних парних елементів масиву array розміру length, що розташовані після якогось елемента масиву та виводить його адресу

2. Аргументи:

1. array – адреса масиву.

2. length – розмір масиву.

3. max_pos – позиція елемента масива.

3. Функція виводить позицію максимального елемента.

8. int * task2(int * array, int * length)

1. Створює новий масив у два рази менший ніж масив array і забирає з масиву елементи з парними індексами у відліку від одного, видаляє array, замінює довжину length та виводить адресу нового масиву.

2. Аргументи:

1. array – адреса масиву.

2. length – адреса звінної, де зберігає розмір масиву.

3. Функція виводить адресу нового масиву

2. У головній функції виконуються такі дії:

1. Налаштовуємо насіння (seed) для генератора випадкових чисел на час виконання програми.

2. Ініціюємо потрібні.

3. Циклічно запитуємо довжину масиву доки не отримаємо число від 1 до 1000

4. Циклічно запитуємо інтервал можливих чисел при генерації доки не отримаємо коректну відповідь.

5. Створюємо масив
6. Заповнюємо масив
7. Показуємо оригінальний масив
8. Циклічно виконуємо:
 1. Запитуємо меню, де вибираємо підзадачу
 2. Залежно від підзадачі можливі такі дії
 1. Перша підзадача
 1. Знаходження максимального числа
 2. Обробити масив за допомогою функції task2
 3. Показуємо отриманий масив
 4. Видаляємо отриманий масив
 2. Друга підзадача
 1. Видалити елементи з непарними елементами
 2. Показуємо отриманий масив
9. Видаляємо масив
10. Виводимо успішне закінчення та виходимо з програми

3. Код:

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

void ign_other(FILE * input) {
    char c = 0;
    while (c!='\n')
        c=fgetc(input);
}

char menu (FILE * input, FILE * output, const char * message, char count) {
    char answer = 0;
    while (answer < 1 || answer > count) {
        printf("%s\n",message);
        answer = getchar()-'0';
        ign_other(input);
        if (answer < 1 || answer > count) {
            system("clear");
            printf("Вводить нужно только цифры от 1 до %hi. Попробуйте
ещё\n",count);
        }
    }
    return answer;
}

int * create_array(int length) {
    int * a = new int[length];
```

```

        return a;
    }

void fill_array(int * array,int length,int min,int max) {
    for (int i=0;i < length;i++) {
        array[i]=rand()%(max-min+1)+min;
    }
}

void show_array(FILE * output,int * array,int length) {
    if (length == 0) {
        fprintf(output,"Массив порожній\n");
        return;
    }
    fprintf(output,"Массив має такі значення:\n");
    for (int i=0;i < length;i++) {
        fprintf(output,"%i ",array[i]);
    }
    fprintf(output,"\n\n");
}

int find_max(int * array, int length) {
    int max=0,i;
    for(i=1;i<length;i++) {
        if(array[max]<array[i])
            max=i;
    }
    return max;
}

int * task1(int * array, int length,int * length_out, int max_pos) {
    if (max_pos>=length) {
        *length_out = 0;
        return new int [0];
    }
    int lengthb=0, *b = new int[length-max_pos],i;
    for (i=max_pos+1;i<length;i++) {
        if (array[i]%2==0 && array[i]>=0) {
            b[lengthb++] = array[i];
        }
    }
    *length_out = lengthb;
    return b;
}

int * task2(int * array, int * length) {
    int *b,l = *length/2,i;
    b = new int[l];
    for (i=0;i<l;i++)
        b[i]=array[i*2+1];
    delete array;
    *length = l;
    return b;
}

int main() {
    srand(time(0));
    int * a,length=0,max=0,intermin=0,intermax=0,answer=0;

```

```

while (length < 1 || length > 1000) {
    printf("Введіть довжину масиву:\n");
    scanf("%i",&length);
    ign_other(stdin);
    if (length < 1 || length > 1000) {
        system("clear");
        printf("Вы ввели неправильну довжину, дозволено тільки ціле число
від 1 до 1000\n");
    }
}
while (intermin >= intermax) {
    printf("Введіть інтервал дозволених значень в масиві:\n");
    scanf("%i%i",&intermin,&intermax);
    ign_other(stdin);
    if (intermin >= intermax) {
        system("clear");
        printf("Вы ввели неправильний інтервал, дозволено тільки цілі
числа, при чому перше число повинно бути менше ніж друге\n");
    }
}
a=create_array(length);
fill_array(a,length,intermin,intermax);
show_array(stdout,a,length);
do {
    answer = menu(stdin,stdout,"Виберіть дії:\n1. Сформувати новий масив з
додатних парних елементів данного масиву, що розташовані після максимального
елемента масиву.\n2. Видалити всі елементи з непарними індексами.\n3. Вихід",3);
    if (answer == 1) {
        max=find_max(a,length);
        printf("Максимальний елемент знаходиться на %i позиції.\n",max);
        if (max == length) printf("Максимальний елемент знаходиться у
кінці масива, тому вихідний масив порожній\n");
        else {
            int * b,lengthb;
            b=task1(a,length,&lengthb,max);
            show_array(stdout,b,lengthb);
            delete b;
        }
    } else if (answer == 2) {
        a=task2(a,&length);
        show_array(stdout,a,length);
    }
} while (answer != 3);
delete a;
return 0;
}

```

4. Результати роботи програми:

Приклад 1:

Введіть довжину масиву:

20

Введіть інтервал дозволених значень в масиві:

-20

20

Масив має такі значення:

8 -19 6 2 15 -3 -4 0 9 9 -13 -19 18 -2 -14 -17 15 -2 20 8

Виберіть дії:

1. Сформувати новий масив з додатних парних елементів данного масиву, що розташовані після максимального елемента масиву.
2. Видалити всі елементи з непарними індексами.
3. Вихід

1

Максимальний елемент знаходиться на 18 позиції.

Масив має такі значення:

8

Виберіть дії:

1. Сформувати новий масив з додатних парних елементів данного масиву, що розташовані після максимального елемента масиву.
2. Видалити всі елементи з непарними індексами.
3. Вихід

2

Масив має такі значення:

-19 2 -3 0 9 -19 -2 -17 -2 8

Виберіть дії:

1. Сформувати новий масив з додатних парних елементів данного масиву, що розташовані після максимального елемента масиву.
2. Видалити всі елементи з непарними індексами.
3. Вихід

2

Масив має такі значення:

2 0 -19 -17 8

Виберіть дії:

1. Сформувати новий масив з додатних парних елементів данного масиву, що розташовані після максимального елемента масиву.
2. Видалити всі елементи з непарними індексами.
3. Вихід

3