

# Звіт з лабораторної роботи за дисципліною "Програмування II" студента групи ПА-17-1 Панасенка Єгора Сергійовича Кафедра комп'ютерних технологій ФПМ, ДНУ, 2017-2018 навч.р.

---

## Приклад роботи програми:

---

```
$ ./main 10 -10 20
```

Виберете функцію:

1. Сложить массив в одно число.
2. Умножить какое-то число на каждый элемент массива.
3. Разделить массив на два массива, где в первом положительные, а во втором отрицательные.
4. Соединить все массивы в один сложением.
0. Выход

```
>>> 1
```

Входящий массив:

```
3 6 17 15 13 15 6 2 -1 -9
```

Сумма массива: 67

Виберете функцію:

1. Сложить массив в одно число.
2. Умножить какое-то число на каждый элемент массива.
3. Разделить массив на два массива, где в первом положительные, а во втором отрицательные.
4. Соединить все массивы в один сложением.
0. Выход

```
>>> 2
```

Входящее число: -8

Входящий массив:

```
-3 10 9 13 6 -10 -4 12 6 1
```

Выходящий массив:

```
24 -80 -72 -104 -48 80 32 -96 -48 -8
```

Виберете функцію:

1. Сложить массив в одно число.
2. Умножить какое-то число на каждый элемент массива.
3. Разделить массив на два массива, где в первом положительные, а во втором отрицательные.
4. Соединить все массивы в один сложением.
0. Выход

```
>>> 3
```

Входящий массив:

```
-2 17 -1 -8 10 -8 3 -3 15 19
```

Массив массив отрицательных чисел:

```
-2 -1 -8 -8 -3
```

Массив массив положительных чисел:

```
17 10 3 15 19
```

Выберете функцию:

1. Сложить массив в одно число.
2. Умножить какое-то число на каждый элемент массива.
3. Разделить массив на два массива, где в первом положительные, а во втором отрицательные.
4. Соединить все массивы в один сложением.
0. Выход

```
>>> 4
```

Входящий первый массив:

```
2 2 8 19 17 3 6 -9 12 -1
```

Входящий второй массив:

```
-7 11 19 4 -3 -2 4 -5 -10 13
```

Выходящий массив:

```
-5 13 27 23 14 1 10 -14 2 12
```

Выберете функцию:

1. Сложить массив в одно число.
  2. Умножить какое-то число на каждый элемент массива.
  3. Разделить массив на два массива, где в первом положительные, а во втором отрицательные.
  4. Соединить все массивы в один сложением.
  0. Выход
- ```
>>> 0
```

## Код:

### Makefile

```
OBJ = main.o functions.o
DEPS = functions.h
LIBS =
CFLAGS = -Wall
CC = gcc
EXTENSION = .c

%.o: %$(EXTENSION) $(DEPS)
    $(CC) -c -o $@ $< $(CFLAGS)

main: $(OBJ)
    $(CC) -o $@ $^ $(CFLAGS) $(LIBS)

.PHONY: clean

clean:
    rm -f *.o *~ core *~ main
```

### functions.h

```
#ifndef FUNCTIONS_H
#define FUNCTIONS_H
#define NUM_SIZE 8
```

```

#include "stdio.h"
#include "stdlib.h"

void ign_other(FILE * input);

int get_number(FILE * input, int * number);

int menu (FILE * input, FILE * output, char * message, int count);

int rnd(int min, int max);

int * generate_array(int n, int min, int max);

void print_array(FILE * input, int * a, int n);

int summarize(int * a, int n);

void multiply_elem(int x, int * a, int n);

void split(int * a, int n, int * b, int *bn, int * c, int * cn);

int * summarize_arrays(int * a, int * b, int n);

#endif

```

## functions.c

```

#include "functions.h"

/* Ignore all data until end of line */
void ign_other(FILE * input) {
    char c = 0;
    while ((c = fgetc(input)) != EOF && c != '\n') {}
}

int get_number(FILE * input, int * number) {
    char c, i = 0, neg = 1, ret = -1; int x = 0;
    while ((c = fgetc(input)) != '\n') {
        if (ret == -1) {
            if (i == 0 && c == '-') neg = -1;
            else if ((i == 0 && c == ' ')
                || (neg == -1 && i == 1 && c == ' ')
                || (c < '0' && c > '9')) ret = 1;
            else if (c == ' ') ret = 0;
            else if (c >= '0' && c <= '9' && i < NUM_SIZE) {
                x = x * 10 + c - '0'; i++;
            }
            else ret = 1;
        }
    }
    if (i == 0 || (neg == -1 && i == 1)) ret = 1;
    else if (ret == -1) ret = 0;
    if (ret == 0) *number = neg * x;
    return ret;
}

```

```

int menu (FILE * input, FILE * output, char * message, int count) {
    // Main menu
    int answer = -1;
    // While answer is incorrect make menu
    while (answer < 0 || answer >= count) {
        // Ask menu
        printf("%s", message);
        // Get answer
        if (get_number(input, &answer) != 0)
            answer = -2;
        // Ask to try again
        if (answer < 0 || answer >= count) {
            system("clear");
            printf(
                "Вводить нужно только цифры от 0 до %hi. Попробуйте ещё\n",
                count - 1);
        }
    }
    return answer;
}

int rnd(int min, int max) {
    return rand() % (max - min) + min;
}

int * generate_array(int n, int min, int max) {
    int * a = malloc(n * sizeof(int)), i;
    if (a == NULL) return NULL;
    for (i = 0; i < n; i++)
        a[i] = rnd(min, max);
    return a;
}

void print_array(FILE * input, int * a, int n) {
    if (n == 0) {
        printf("Массив пустой");
        return;
    }
    for (int i = 0; i < n; i++)
        printf("%i ", a[i]);
    printf("\n");
}

int summarize(int * a, int n) {
    int num = 0, i;
    for (i = 0; i < n; i++)
        num += a[i];
    return num;
}

void multiply_elem(int x, int * a, int n) {
    for (int i = 0; i < n; i++) a[i] *= x;
}

void split(int * a, int n, int * b, int *bn, int * c, int *cn) {
    *bn = 0; *cn = 0;
    for (int i = 0; i < n; i++)
        if (a[i] < 0) b[(*bn)++] = a[i];

```

```

        else c[(*cn)++] = a[i];
    }

    int * summarize_arrays(int * a, int * b, int n) {
        int * c = malloc(n * sizeof(int)), i;
        if (a == NULL) return NULL;
        for (i = 0; i < n; i++) c[i] = a[i] + b[i];
        return c;
    }

```

## main.c

```

#include "functions.h"
#include "stdlib.h"
#include "stdio.h"

int main(int argc, char * argv[]) {
    if (argc < 4) printf("Слишком мало аргументов\n");
    if (argc > 4) printf("Слишком много аргументов\n");
    if (argc != 4) return 1;
    int n, bn, cn, min, max, *a = NULL, *b, *c, tmp;
    n = atoi(argv[1]); min = atoi(argv[2]); max = atoi(argv[3]);
    if (n <= 0) {
        printf
            ("Размер массива не может быть нулевым или отрицательным");
        return 1;
    }
    if (min > max) {
        tmp = min; max = min; min = tmp;
    }
    int answer = -1;
    while (answer) {
        answer = menu(stdin, stdout, "Выберете функцию:\n\
1. Сложить массив в одно число.\n\
2. Умножить какое-то число на каждый элемент массива.\n\
3. Разделить массив на два массива, где в первом положительные, \n\
   а во втором отрицательные.\n\
4. Соединить все массивы в один сложением.\n\
0. Выход\n\
>>> ", 5);

        //~ system("clear");
        switch(answer) {
            case 1:
                a = generate_array(n, min, max);
                if (a == NULL) {
                    printf("Не удалось выделить память\n");
                    break;
                }
                printf("Входящий массив:\n");
                print_array(stdin, a, n);
                printf("Сумма массива: %i\n", summarize(a, n));
                free(a);
                break;
            case 2:
                tmp = rnd(min, max);
                a = generate_array(n, min, max);

```

```

        if (a == NULL) {printf("Не удалось выделить память\n");
                        break;
        }
        printf("Входящее число: %i\n", tmp);
        printf("Входящий массив:\n");
        print_array(stdin, a, n);
        multiply_elem(tmp, a, n);
        printf("Выходящий массив:\n");
        print_array(stdin, a, n);
        free(a);
        break;
    case 3:
        a = generate_array(n, min, max);
        b = malloc(n * sizeof(int));
        c = malloc(n * sizeof(int));
        if (a == NULL || b == NULL || c == NULL) {
            printf("Не удалось выделить память\n");
            free(a);free(b);free(c);break;
        }
        printf("Входящий массив:\n");
        print_array(stdin, a, n);
        split(a, n, b, &bn, c, &cn);
        printf("Массив массив отрицательных чисел:\n");
        print_array(stdin, b, bn);
        printf("Массив массив положительных чисел:\n");
        print_array(stdin, c, cn);
        if (!a) free(a);
        if (!b) free(a);
        break;
    case 4:
        a = generate_array(n, min, max);
        b = generate_array(n, min, max);
        if (a == NULL || b == NULL) {
            printf("Не удалось выделить память\n");
            free(a);free(b);break;
        }
        printf("Входящий первый массив:\n");
        print_array(stdin, a, n);
        printf("Входящий второй массив:\n");
        print_array(stdin, b, n);
        c = summarize_arrays(a, b, n);
        if (c == NULL) {
            printf("Не удалось выделить память\n");
            break;
        }
        printf("Выходящий массив:\n");
        print_array(stdin, c, n);
        free(a);free(b);free(c);
        break;
    };
    if (answer) ign_other(stdin);
    //~ system("clear");
}
}

```