**Курс «Язык программирования C»**

Тема: Препроцессор

Домашнее задание

**Задание:**

С помощью директивы #define написать следующие макросы:

* Нахождение меньшего из двух чисел;
* Нахождение большего из двух чисел;
* Возведение числа в квадрат;
* Возведение числа в степень;
* Проверка числа на четность;
* Проверка числа на нечетность;

#include <math.h>

#include <stdio.h>

#define min(\_number\_one, \_number\_two) ((\_number\_one < \_number\_two) ? \_number\_one : \_number\_two)

#define max(\_number\_one, \_number\_two) ((\_number\_one > \_number\_two) ? \_number\_one : \_number\_two)

#define squaring(number) number \* number

#define exponentiation(number, power) (pow(number, power))

#define EvenNumber(number) number % 2 == 0

#define OddNumber(number) number % 2 != 0

int main() {

short choise = 0;

int \_number\_one = 0, \_number\_two = 0;

do {

printf("%s", "\n\tmenu\n");

printf("%s", "1. finding the smaller of the two numbers.\n");

printf("%s", "2. finding the larger of the two numbers.\n");

printf("%s", "3. construction of the square.\n");

printf("%s", "4. construction of the power.\n");

printf("%s", "5. check the number for parity.\n");

printf("%s", "6. check the number for odd.\n");

printf("%s", "select menu item: ");

scanf\_s("%d", &choise);

switch (choise) {

case 1:

printf("%s", "enter a number: ");

scanf\_s("%d", &\_number\_one);

printf("%s", "enter a number: ");

scanf\_s("%d", &\_number\_two);

printf("%s%d%c", "the smaller of the two numbers: ", min(\_number\_one, \_number\_two), '\n');

break;

case 2:

printf("%s", "enter a number: ");

scanf\_s("%d", &\_number\_one);

printf("%s", "enter a number: ");

scanf\_s("%d", &\_number\_two);

printf("%s%d%c", "the greater of the two numbers: ", max(\_number\_one, \_number\_two), '\n');

break;

case 3:

printf("%s", "enter a number: ");

scanf\_s("%d", &\_number\_one);

printf("%s%d%c", "square root of a number: ", squaring(\_number\_one), '\n');

break;

case 4:

printf("%s", "enter a number: ");

scanf\_s("%d", &\_number\_one);

printf("%s", "enter a number degree: ");

scanf\_s("%d", &\_number\_two);

\_number\_one = exponentiation(\_number\_one, \_number\_two);

printf("%s%d%s%d%s%d%c", "the number ", \_number\_one, " in degree ", \_number\_two, " is equal to ", \_number\_one, '\n');

break;

case 5:

printf("%s", "enter a number: ");

scanf\_s("%d", &\_number\_one);

if (EvenNumber(\_number\_one))

printf("%s", "the number is even\n");

else

printf("%s", "the number is odd.\n");

break;

case 6:

printf("%s", "enter a number: ");

scanf\_s("%d", &\_number\_one);

if (OddNumber(\_number\_one))

printf("%s", "the number is odd\n");

else

printf("%s", "the number is even.\n");

break;

default:

printf("%s", "no menu item selected, select another menu item.\n");

break;

}

printf("%s", "retry the operation(1. yes/0. no)?: ");

scanf\_s("%d", &choise);

} while (choise);

return 0;

}

**Создать проект в котором :**1. создать три файла: ( function.h , function.cpp , prog.cpp )  
2. в файле function.cpp необходимо написать следующие функции для работы с массивом данных:

а) функцию для заполнения массива случайными значениями ;  
б) функцию для вывода значений массива на консоль ;  
в) функцию для поиска минимального элемента;  
г) функцию для поиска максимального элемента;  
д) функцию для сортировки ;  
е) функцию для редактирования значения массива.  
Данные функции необходимо написать для работы с массивом целых, действительных и символьных значений. (Для каждого типа написать отдельную функцию) .  
3. в файле function.h нужно:  
а) описать прототипы всех функций;  
4. в файле prog.cpp нужно :

a) сделать проверку определена ли константа, указывающая на тип данных ( #ifdef INTEGER )  
б) в функции main вызвать все функции .

Function.h

#pragma once

#include <stdlib.h>

#include <stdio.h>

#include <time.h>

// условная конструкция для обобщения имен функций

#ifdef NINTEGER

#define \_data\_type int

#define \_fill\_array \_fill\_int\_array\_random\_numbers

#define \_output\_array \_output\_int\_array

#define \_finding\_minimum\_item search\_int\_minimum\_element

#define \_finding\_maximum\_item search\_int\_maximum\_element

#define \_sorting\_array \_sorting\_int\_array

#define \_change\_value\_array \_change\_value\_int\_array

#endif

#ifdef NDOUBLE

#define \_data\_type double

#define \_fill\_array \_fill\_double\_array\_random\_numbers

#define \_output\_array \_output\_double\_array

#define \_finding\_minimum\_item search\_double\_minimum\_element

#define \_finding\_maximum\_item search\_double\_maximum\_element

#define \_sorting\_array \_sorting\_double\_array

#define \_change\_value\_array \_change\_value\_double\_array

#endif

#ifdef NCHAR

#define \_data\_type char

#define \_fill\_array \_fill\_char\_array\_random\_numbers

#define \_output\_array \_output\_char\_array

#define \_finding\_minimum\_item search\_char\_minimum\_element

#define \_finding\_maximum\_item search\_char\_maximum\_element

#define \_sorting\_array \_sorting\_char\_array

#define \_change\_value\_array \_change\_value\_char\_array

#endif

// функции заполнения массива

void \_fill\_int\_array\_random\_numbers(int\* array, size\_t size);

void \_fill\_double\_array\_random\_numbers(double\* array, size\_t size);

void \_fill\_char\_array\_random\_numbers(char\* array, size\_t size);

// функции вывода массива на консоль

void \_output\_int\_array(const int\* array, size\_t size);

void \_output\_double\_array(const double\* array, size\_t size);

void \_output\_char\_array(const char\* array, size\_t size);

// функции поиска минимального элемента

int search\_int\_minimum\_element(const int\* array, size\_t size);

double search\_double\_minimum\_element(const double\* array, size\_t size);

char search\_char\_minimum\_element(const char\* array, size\_t size);

// функции поиска максимального элемента

int search\_int\_maximum\_element(const int\* array, size\_t size);

double search\_double\_maximum\_element(const double\* array, size\_t size);

char search\_char\_maximum\_element(const char\* array, size\_t size);

// функции сортировки массива

void \_sorting\_int\_array(int\* array, size\_t size);

void \_sorting\_double\_array(double\* array, size\_t size);

void \_sorting\_char\_array(char\* array, size\_t size);

// функции редактирования значения массива

void \_change\_value\_int\_array(int\* array, size\_t size, const int index, const int value);

void \_change\_value\_double\_array(double\* array, size\_t size, const int index, const double value);

void \_change\_value\_char\_array(char\* array, size\_t size, const int index, const char value);

Function.cpp

#include "Functions.h"

// функции заполнения массива

void \_fill\_int\_array\_random\_numbers(int\* array, size\_t size) {

srand(static\_cast<unsigned int>(time(NULL)));

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

array[i] = rand() % 21;

}

void \_fill\_double\_array\_random\_numbers(double\* array, size\_t size) {

srand(static\_cast<unsigned int>(time(NULL)));

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

array[i] = rand() / 32767.0 \* 21;

}

void \_fill\_char\_array\_random\_numbers(char\* array, size\_t size) {

srand(static\_cast<unsigned int>(time(NULL)));

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

array[i] = static\_cast<char>(rand() % 256);

}

// функции вывода массива на консоль

void \_output\_int\_array(const int\* array, size\_t size) {

printf("%s", "array: ");

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

printf("%c%d", '|', array[i]);

printf("%c%c", '|', '\n');

}

void \_output\_double\_array(const double\* array, size\_t size) {

printf("%s", "array: ");

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

printf("%c%f", '|', array[i]);

printf("%c%c", '|', '\n');

}

void \_output\_char\_array(const char\* array, size\_t size) {

printf("%s", "array: ");

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

printf("%c%c", '|', array[i]);

printf("%c%c", '|', '\n');

}

// функции поиска минимального элемента

int search\_int\_minimum\_element(const int\* array, size\_t size) {

int minimum = array[0];

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

if (array[i] < minimum)

minimum = array[i];

return minimum;

}

double search\_double\_minimum\_element(const double\* array, size\_t size) {

double minimum = array[0];

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

if (array[i] < minimum)

minimum = array[i];

return minimum;

}

char search\_char\_minimum\_element(const char\* array, size\_t size) {

char minimum = array[0];

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

if (array[i] < minimum)

minimum = array[i];

return minimum;

}

// функции поиска максимального элемента

int search\_int\_maximum\_element(const int\* array, size\_t size) {

int maximum = array[0];

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

if (array[i] > maximum)

maximum = array[i];

return maximum;

}

double search\_double\_maximum\_element(const double\* array, size\_t size) {

double maximum = array[0];

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

if (array[i] > maximum)

maximum = array[i];

return maximum;

}

char search\_char\_maximum\_element(const char\* array, size\_t size) {

char maximum = array[0];

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

if (array[i] > maximum)

maximum = array[i];

return maximum;

}

// функции сортировки массива

void \_sorting\_int\_array(int\* array, size\_t size) {

int \_temporary\_variable;

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

for (unsigned int j = size - 1; j > i; j--) {

if (array[j] < array[i]) {

\_temporary\_variable = array[i];

array[i] = array[j];

array[j] = \_temporary\_variable;

}

}

}

}

void \_sorting\_double\_array(double\* array, size\_t size) {

double \_temporary\_variable;

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

for (unsigned int j = size - 1; j > i; j--) {

if (array[j] < array[i]) {

\_temporary\_variable = array[i];

array[i] = array[j];

array[j] = \_temporary\_variable;

}

}

}

}

void \_sorting\_char\_array(char\* array, size\_t size) {

char \_temporary\_variable;

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

for (unsigned int j = size - 1; j > i; j--) {

if (array[j] < array[i]) {

\_temporary\_variable = array[i];

array[i] = array[j];

array[j] = \_temporary\_variable;

}

}

}

}

// функции редактирования значения массива

void \_change\_value\_int\_array(int\* array, size\_t size, const int index, const int value) {

(index > 0 && index < size) ? array[index] = value : printf("%s", "there is no index entered in the array, enter the correct index sequence number.\n");

}

void \_change\_value\_double\_array(double\* array, size\_t size, const int index, const double value) {

(index > 0 && index < size) ? array[index] = value : printf("%s", "there is no index entered in the array, enter the correct index sequence number.\n");

}

void \_change\_value\_char\_array(char\* array, size\_t size, const int index, const char value) {

(index > 0 && index < size) ? array[index] = value : printf("%s", "there is no index entered in the array, enter the correct index sequence number.\n");

}

Main.cpp

#define NINTEGER

#define LENGTH 10

#include "Functions.h"

#include <Windows.h>

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int choise = 0;

int index = 0;

int value = 0;

char symbol = '\0';

\_data\_type array[LENGTH];

do {

printf("%s", "\tменю\n");

printf("%s", "1. заполнение массива.\n");

printf("%s", "2. поиск минимального элемента массива.\n");

printf("%s", "3. поиск максимального элемента массива.\n");

printf("%s", "4. сортировка массива.\n");

printf("%s", "5. изменение значения массива.\n");

printf("%s", "6. вывод массива на консоль.\n");

printf("%s", "выберите пункт меню: ");

scanf\_s("%d", &choise);

switch (choise) {

case 1:

\_fill\_array(array, LENGTH);

break;

case 2:

printf("%s%d%c", "минимальное значение в массиве: ", \_finding\_minimum\_item(array, LENGTH), '\n');

break;

case 3:

printf("%s%d%c", "максимальное значение в массиве: ", \_finding\_maximum\_item(array, LENGTH), '\n');

break;

case 4:

\_sorting\_array(array, LENGTH);

break;

case 5:

printf("%s", "введите порядковый номер значения в массиве: ");

scanf\_s("%d", &index);

printf("%s", "введите новое значение: ");

scanf\_s("%d", &value);

\_change\_value\_array(array, LENGTH, index, value);

break;

case 6:

\_output\_array(array, 10);

break;

default:

printf("%s", "в меню нет пункта под выбраным номером, введите другой номер.\n");

break;

}

printf("%s", "повторить операцию(y/n)? ");

scanf\_s("%c", &symbol);

scanf\_s("%c", &symbol);

} while (symbol != 'n' && symbol != 'N');

return 0;

}