#include<iostream>

using namespace std;

int main()

{

int n;

std::cout << "N=";

std::cin >> n;

float \*x = new float[n];

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

std::cout << "x(" << i << ") = ";

std::cin >> x[i];

}

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

std::cout << "x(" << i << ") = "

<< x[i] << "\n";

}

return 0;

}

1.

#include<iostream>

using namespace std;

int main()

{

int n;

std::cout << "N = ";

std::cin >> n;

double \*x = new double[n];

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

std::cout << "x(" << i << ") = ";

std::cin >> x[i];

}

double a, b;

std::cout << "A = ";

std::cin >> a;

std::cout << "B = ";

std::cin >> b;

int k = 0;

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

if ((static\_cast<int>(x[i]) % 2 != 0) && ((x[i] < a) or (x[i] > b))) {

k += 1;

}

}

std::cout << "Количество нечетных элементов массива, не попадающих в интервал [" << a << ";" << b << "]: "

<< k << "\n";

std::cout << "Положительные элементы массива с чётными индексами: \n";

k = 0;

for (int i=0; i<n; i += 2){

if (x[i] > 0) {

k += 1;

std::cout << x[i] << ", ";

}

}

if (k == 0) {

std::cout << "Отсутствуют";

}

return 0;

}

3.

#include <iostream>

using namespace std;

int main() {

int n;

std::cout << "N = ";

std::cin >> n;

float \*x = new float[n];

float \*y = new float[n];

float \*z = new float[n];

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

std::cout << "x(" << i << ") = ";

std::cin >> x[i];

std::cout << "y(" << i << ") = ";

std::cin >> y[i];

}

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

if (x[i] < y[i]) {

z[i] = x[i];

} else {

z[i] = y[i];

}

}

std::cout << "---Вывод массива Zk---" << "\n";

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

std::cout << "z(" << i << ") = " << z[i] << "\n";

}

std::cout << "---Cортировка массива Xn по возрастанию---" <<"\n";

while (true) {

bool fsort = true;

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

if (!(x[i] <= x[i + 1])) {

double s = x[i];

x[i] = x[i + 1];

x[i+1] = s;

fsort = false;

}

}

if (fsort) {

break;

}

}

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

std::cout << "x(" << i << ") = " << x[i] << "\n";

}

return 0;

}

2.

#include<iostream>

using namespace std;

int main() {

int n;

std::cout << "N = ";

std::cin >> n;

float \*y = new float[n];

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

std::cout << "y(" << i << ") = ";

std::cin >> y[i];

}

int k = 0;

for (int i = 1; i < n; i += 2) {

if (static\_cast<int>(y[i]) % 2 == 0) {

k += 1;

}

}

std::cout << "Количество чётных элементов с нечётными номерами: " << k << "\n";

int j = 0;

std::cout << "---Удаление нулевых элементов--- \n";

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

if (y[i] != 0) {

y[j] = y[i];

j += 1;

}

}

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

std::cout << "y(" << i << ") = " << y[i] << "\n";

}

if (j == n) {

std::cout << "Нулевые элементы не обнаружены \n";

}

std::cout << "---Замена предпоследнего отрицательного элемента нулём--- \n";

k = 0;

for (int i = j - 1; i >= 0; i--) {

if (y[i] < 0) {

k += 1;

if (k == 2) {

y[i] = 0;

break;

}

}

}

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

std::cout << "y(" << i << ") = " << y[i] << "\n";

}

if (k < 2) {

std::cout << "Нет подходящего элемента, замен не производилось";

}

return 0;

}

Project : Prog4 Пример 4 Мигающий светодиод

Chip type : ATtiny2313

Clock frequency 4, 000000 MHz

#include <tiny2313.h>

#include <delay.h>

void main(void)

PORTB=0xFF;

DDRB=0xFF; // Инициализация порта

PORTD=0x7F;

DDRD=0x00; // Инициализация порта D

ACSR=0x80; // Инициализация аналогового компаратора

while (1)

{

if (PIND.0==1)

{PORTB.0=1; }

else

{

PORTB. 0=1;

delay\_ms(200);

PORTB. 0=0;

delay\_ms(200);

PORTB. 0=1;

delay\_ms(200);

PORTB. 0=0;

delay\_ms(600);

PORTB. 0=1;

delay\_ms(200);

PORTB. 0=0;

delay\_ms(600);

}

};

}

; Программа для ATTiny2313, которая мигает светодиодом

; в соответствии с азбукой Морзе для имени "Дарья"

.include "tn2313def.inc"

; Определяем временные задержки

.equ DOT\_DELAY = 300 ; Длительность точки (в миллисекундах)

.equ DASH\_DELAY = 2100 ; Длительность тире (в миллисекундах)

.equ INTER\_ELEMENT\_DELAY = 300 ; Задержка между элементами

.equ LETTER\_SPACE\_DELAY = 2100 ; Задержка между буквами

; Инициализация порта

.org 0x00

rjmp START

START:

; Настройка порта B0 как выход

ldi r16, (1 << PB0)

out DDRB, r16

MAIN\_LOOP:

; Буква Д: -..

rjmp DASH

rjmp DOT

rjmp DOT

rjmp LETTER\_SPACE

; Буква А: .-

rjmp DOT

rjmp DASH

rjmp LETTER\_SPACE

; Буква Р: .-.

rjmp DOT

rjmp DASH

rjmp DOT

rjmp LETTER\_SPACE

; Буква Ь: -..

rjmp DASH

rjmp DOT

rjmp DOT

rjmp LETTER\_SPACE

; Буква Я: .-.

rjmp DOT

rjmp DASH

rjmp DOT

; Пауза перед повторением

rjmp LETTER\_SPACE

rjmp MAIN\_LOOP

; Подпрограмма для точки

DOT:

sbi PORTB, PB0 ; Включаем светодиод

ldi r20, DOT\_DELAY / 10 ; Делим на 10 для циклов (примерно)

DELAY\_DOT:

ldi r21, 255 ; Внутренний счетчик

DELAY\_DOT\_LOOP:

dec r21 ; Уменьшаем счетчик

brne DELAY\_DOT\_LOOP ; Если не ноль, продолжаем задержку

dec r20 ; Уменьшаем внешний счетчик

brne DELAY\_DOT ; Если не ноль, повторяем задержку

cbi PORTB, PB0 ; Выключаем светодиод

ldi r20, INTER\_ELEMENT\_DELAY / 10 ; Задержка между элементами

DELAY\_INTER\_ELEMENT:

ldi r21, 255 ; Внутренний счетчик

DELAY\_INTER\_ELEMENT\_LOOP:

dec r21 ; Уменьшаем счетчик

brne DELAY\_INTER\_ELEMENT\_LOOP ; Если не ноль, продолжаем задержку

dec r20 ; Уменьшаем внешний счетчик

brne DELAY\_INTER\_ELEMENT ; Если не ноль, повторяем задержку

ret

; Подпрограмма для тире

DASH:

sbi PORTB, PB0 ; Включаем светодиод

ldi r20, DASH\_DELAY / 10 ; Делим на 10 для циклов (примерно)

DELAY\_DASH:

ldi r21, 255 ; Внутренний счетчик

DELAY\_DASH\_LOOP:

dec r21 ; Уменьшаем счетчик

brne DELAY\_DASH\_LOOP ; Если не ноль, продолжаем задержку

dec r20 ; Уменьшаем внешний счетчик

brne DELAY\_DASH ; Если не ноль, повторяем задержку

cbi PORTB, PB0 ; Выключаем светодиод

ldi r20, INTER\_ELEMENT\_DELAY / 10 ; Задержка между элементами

DELAY\_INTER\_ELEMENT\_AFTER\_DASH:

ldi r21, 255 ; Внутренний счетчик

DELAY\_INTER\_ELEMENT\_AFTER\_DASH\_LOOP:

dec r21 ; Уменьшаем счетчик

brne DELAY\_INTER\_ELEMENT\_AFTER\_DASH\_LOOP ; Если не ноль, продолжаем задержку

dec r20 ; Уменьшаем внешний счетчик

brne DELAY\_INTER\_ELEMENT\_AFTER\_DASH ; Если не ноль, повторяем задержку

ret

; Задержка между буквами

LETTER\_SPACE:

ldi r20, LETTER\_SPACE\_DELAY / 10 ; Делим на 10 для циклов (примерно)

DELAY\_LETTER\_SPACE:

ldi r21, 255 ; Внутренний счетчик

DELAY\_LETTER\_SPACE\_LOOP:

dec r21 ; Уменьшаем счетчик

brne DELAY\_LETTER\_SPACE\_LOOP ; Если не ноль, продолжаем задержку

dec r20 ; Уменьшаем внешний счетчик

brne DELAY\_LETTER\_SPACE ; Если не ноль, повторяем задержку

ret

.org 0x1FF ; Конец программы

;---------------------------------------------------------------------------- Псевдокоманды управления

format PE console 4.0

entry start

include 'C:\fasmw17330\INCLUDE\win32ax.inc' ;присоединение файла win32ax.inc

section '.data' data readable writable

;----------------------------------------------------------------------------- Текст

rez db 13,10,'mass[%d,%d] = %d',0

;----------------------------------------------------------------------------- Переменные

n dd 4

m dd 3

mass dd 5, 6, 3, -7, 8, 10, 0, 2, 1, 12, -8, 4

buf rd 255

;----------------------------------------------------------------------------- Начало кода

section '.code' code readable executable

start:

mov eax, 10 ;ЕАХ присваивается значение 10

mov ebx, 7 ;ЕВХ присваивается значение 7

imul ebx, 4 ;умножение на 4

imul eax, 4

;ЕСХ присваивается значение 10 элемента

mov ecx, [mass + eax] ;ЕИХ присваивается значение 7 элемента

mov edx, [mass + ebx] ;10 элементу присваивается значение 7 элемента

mov [mass + eax], edx ;7 элементу присваивается значение 10 элемента

mov [mass + ebx], ecx

mov esi, 0 ;перевод в нули

mov ebx, 0

mov edi, 0

n2:

mov edi, 0

n3:

mov ebx, [m] ;создание смещения

imul ebx, esi

add ebx, edi

imul ebx, 4

cinvoke printf, rez, esi, edi, [mass+ebx] ;вывод массива

inc edi

cmp edi, [m]

jne n3 ;если не равно, то n3

inc esi

cmp esi, [n]

jne n2 ;если не равно, то n2

invoke sleep, 10000

invoke exit, 0

section '.idata' import data readable writeable

library user32,'USER32.DLL', msvcrt, 'MSVCRT.DLL', kernel32,'KERNEL32.DLL'

import kernel32,sleep,'Sleep'

import msvcrt, puts,'puts', scanf,'scanf', printf,'printf', exit,'exit'