**Языки программирования**

**Лекции**

**06.09.2023**

**Лекция 1**

1. Компилятор (msbuid, GCC, clang)

2. IDE

VS Code

Microsoft Visual Studio

Code :: Blocks

*Hello world*

*l1.c*

1. #include <stdio.h> - подключаем библиотеку
2. int main() - вводим функцию main
3. {
4. printf(“Hello World”); - printf для вывода текста
5. return 0; - в случае ошибки сообщает о некорректной работе
6. }

l1.c –> l1.exe

**Классы лексем**

1. Идентификаторы
2. Ключевые слова
3. Константы
4. Строковые константы
5. Операторы
6. Прочие разделители (напр. фигурные скобки)

**Комментарии**

1. //однострочный комментарий
2. /\*многострочный

комментарий

\*/

\n – перевод на новую строку print f («Hello /n World»);

\t – табуляция

\\ - вывести «\» как символ

**Как создать переменную:** тип имя\_переменной;

тип список\_переменных;

**Типы переменных**

1. int – целые значения
2. float – дробные
3. char – символьные

5, -10…

g = b1 / b0

b6 = b0 \* q^5

#include <stdio.h>

#include <math.h>

int main()

{

float b0 = 5;

float b1 = -10;

float q, b6;

q = b1 / b0;

b6 = b0 \* pow(q, 5) – ф-ция pow - возведение в степень (q в 5 степени)

printf(“6-й член прогрессии %4.2f, %4.2f равен %4.2f\n”, b0, b1, b6);

return 0;

}

%d - целый

%f - дробный

%c - символ

%4.2f – вывести 4 знака до точки, 2 после (написать можно и %.2 но строки будут не друг под другом)

на место %4.2f выводится b0, b1, b6 последовательно

#include <stdio.h>

int main()

{

int a = 5;

float b = 1.23;

char c = x;

printf(“%d”, a);

printf(“%f”, b);

printf(“%c”, c);

return 0;

}

**13.09.2023**

**Лекция 2**

scanf (“­bla%d”, …)

int a;

scanf(“%d”, &a);

int c, d;

char c1, d1;

scanf(“%d, %d”, &c, &d);

scanf(“%c %\*c %c”, &c1, &d1);

c1 = getch()

c1 = getche()

c1 = getchar()

int x;

x = 10;

x = x+5;

x += 5; - к переменной прибавится 5

x += 1;

x ++; - прибавить 1

int x, y;

x = 10;

y = x++;

// y = 10, x = 11 //

++x – вначале увеличение, потом использование

x++ - использование, потом увеличение

Остаток от деления

int x, y, c;

x = 5;

y = 2;

c = x%y;

// c = 1 //

int – 4 байта (2^32)

float – 4-8 байт short int – уменьшает в 2 раза

char – 1-2 байта long int – увеличивает в 2 раза

unsigned int

&& - и

|| - или

! – «не»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| a | b | a&&b | a||b | !a |
| 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 |

**if**

if (выражение)

{

действия при истинности выражения;

}

else

{

в противном случае;

}

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

int main()

{

int magic;

int guess;

*srand(time(NULL));*

magic = rand()%6; - рандом число от 0 до 5

printf(“Введите число: ”);

scanf(“%d”, &guess);

if(magic == guess)

{

printf(“Вы угадали!”);

}

else

{

printf(“Вы проиграли! \n”);

printf(“Было %d, вы написали %d \n”, magic, guess);

}

return 0;

}

if(guess > magic)

{

printf(“Слишком много”);

}

else

{

printf(“Слишком мало”);

}

printf(“Ошибка! Вы ввели %d, а надо %d”, guess, magic);

}

if((guess != magic) && (guess > magic))

{

printf(“Ошибка, слишком много, надо %d”, magic);

}

else

{

if((guess != magic) && (guess < magic))

{

printf(“Ошибка, слишком мало, надо %d”, magic)

}

if(guess == magic)

{

printf(“Угадали!”);

}

?

int x, y;

x = 10;

y = x>9 ? 100:200;

/\* if (x>9)

y = 100

else

y = 200

\*/

switch(num)

{

case 1: printf(“Январь”);

break;

case 2: printf(“Февраль”);

break;

case 12 …

default: printf(“Вы глупый”);

}

**20.09.2023**

**Лекция 3**

**Оформление отчета**

1. Теоретические сведения (примерно 1 стр.)
2. Задание
3. Исх. код
4. Скриншоты работы
5. Схема алгоритма
6. Вывод

**Схема алгоритма**

Начало

х – ввод/вывод

х = х+4 – операции

х > 4 - ветвление

Пример:

#include <stdio.h>

int main() {

int m;

printf(“Введите номер”);

scanf(“%d”, &m);

switch(m) {

case 1: case 2: case 12:

printf(“Зима”);

break;

default: printf(“Ошибка”);

break;

}

return 0;

}

**Циклы**

1. for (инициализация; условие; приращение)

{

тело цикла;

}

Программа, выводящая все числа от 1 до 100

#include <stdio.h>

int main() {

int i;

for (i=1; i<=100; i++) {

printf("%d; ", i);

}

return 0;

}

int i = 1

for (i; i<=100; ) {

printf(“%d; “, i);

i++;

}

continue

for (int i = 1 ; ;) {

if (i>100) break;

printf(“%d; “, i);

i++;

}

Программа, которая считает сумму введенных пользователем чисел

#include <stdio.h>

int main() {

    int t, S;

   for (printf("Введите число: "), S = 0; scanf("%d", &t), t; S += t, printf("еще "));

   printf("Сумма = %d", S);

   return 0;

}

1. while

while (условие) {

операторы;

}

#include <stdio.h>

int main() {

    int t, S;

    S=0;

    printf("Введите число: ");

    scanf("%d", &t);

    while(t != 0) // while(t)

{

    S += t;

    printf("еще ");

    scanf("%d", &t);

}

printf("Сумма = %d", S);

return 0;

}

#include <stdio.h>

int main() {

    int t, S;

    S=0;

    printf("Введите число: ");

    scanf("%d", &t);

    while(1) {

    S += t;

    printf("еще ");

    scanf("%d", &t);

    if(t == 0) break;

    }

    printf("Сумма = %d", S);

return 0;

}

1. do while

do {

тело

}

while (условие)

#include <stdio.h>

int main() {

    int t, S;

    S = 0;

    printf("Введите число: ");

    do {

        scanf("%d", &t);

        S += t;

    printf("еще ");

    }

    while (t!= 0);

    printf("Сумма = %d", S);

    return 0;

}

**27.09.2023**

**Лекция 4**

// программа выводит число в обратном порядке

#include <stdio.h>

int main() {

    int number;

    printf("Введите число: ");

    scanf("%d", &number);

    while(number) { // number не равно 0

        printf("%d",number%10);

        number /= 10;

    }

    return 0;

}

**Массивы**

тип имя\_массива [кол-во элементов];

int mas [100];

float mas2 [100];

char mas3 [10];

*Первый элемент в массиве имеет индекс 0.*

#include <stdio.h>

int main() {

    int ar [100]; // от 0 до 99

    int i;

    for(i = 0; i < 100; i++) {

        ar[i] = i;

    }

    for(i = 0; i < 100; i++) {

    printf("%d, ", ar[i]);

}

printf("\n");

}

#include <stdio.h>

int main() {

    int ar [1000];

    int N;

    int i;

    printf("Введите количество элементов: ");

    scanf("%d", &N);

    for(i = 0; i < N; i++) {

        printf("Введите %d-й элемент: ", i);

        scanf("%d", &ar[i]);

    }

    return 0;

}

#include <stdio.h>

int main() {

    int ar [1000];

    int N;

    int x [] = {1, 2, 3, 5, 8, 11, 23, 100, 9, 505}; // массив с пустым значением

    char r [] = "Я люблю C";

    char t [] = {'Я', ' ', 'л', 'ю'};

    return 0;

}

#include <stdio.h>

int main() {

    int mas [3] [4]; // двухмерный массив с 3 строками и 4 столбцами

    int i, j;

    for(i = 0; i < 3; i++) {

        for(j = 0; j < 4; j++) {

            printf("Введите [%d] [%d] эл-т: ", i, j);

            scanf("%d", &mas[i] [j]);

        }

    }

printf("\n");

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

    for(j = 0; j < 4; j++) {

        printf("%4d", mas [i] [j]);

    }

    printf("\n");

}

}

**04.10.23**

**Лекция 5.**

**Функции**

**Задать функцию:**

тип\_возвр-зн-я имя (аргументы) {

тело

}

*void* – функция, которая ничего не возвращает обратно после ее использования

**Аргументы** – тип1 имя1, тип2 имя2

#include <stdio.h>

void reverse(int number) {

while(number) {

printf("%d", number%10);

number /= 10;

}

}

 int main() {

    int x;

    printf("Введите число: ", &x);

    scanf("%d", &x);

    reverse(x);

    printf("\n");

    reverse(12345);

    printf("\n");

    reverse(x);

    return 0;

 }

#include <stdio.h>

int minimum(int x, int y) {

    return (x<y)?x:y; //  если х меньше y подставляется значение х, если больше то y

}

    /\* if(x<y)

    return x;

    else

    return y; \*/

    int main() {

        int x, y;

        printf("Введите число: ");

        scanf("%d %d", &x, &y);

        printf("%d\n", minimum(x,y));

        printf("%d\n", minimum(5, -10));

        return 0;

    }

#include <stdio.h>

int global; // переменную можно использовать в любой функции

int minimum(int x, int y) {

    return (x<y)?x:y; //  если х меньше y подставляется значение х, если больше то y

}

    /\* if(x<y)

    return x;

    else

    return y; \*/

    int main() {

        int x, y;

        printf("Введите число: ");

        scanf("%d %d", &x, &y);

        printf("%d\n", minimum(x,y));

        printf("%d\n", minimum(5, -10));

        return 0;

    }

#include <stdio.h>

int find\_number (int number)

{

    int mas [] = {1, -10, 22, 13, -4, 6, 15};

    int N = 7, i;

    for(i = 0; i < N; i++) {

        if(mas[i] == number)

        return i; // exit(0) завершает выполнения программы

    }

    return -1;

}

int main() {

    int x, num;

    printf("Введите иcкомое число: ");

    scanf("%d", &x);

    if(num = find\_number(x) == -1) {

        printf("Число не найдено");

    }

    else {

        printf("Позиция в массиве: %d", num);

    }

}

**11.10.2023**

**Лекция 6**

**Рекурсивные функции**

#include <stdio.h>

int fact1(int n) {

    int t;

    int answer;

    for(t=answer=1; t<=n; answer \*=t, t+1) {

        return answer;

    }

}

// рекурсивный вариант:

int fact2(int n) {

    if(n==0) return 1;

    else

    return fact2(n-1)\*n;

}

#include <stdio.h>

#include <stdlib.h>

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

    if(argc != 2) {

    printf("Вы забыли указать имя");

    return 1;

    }

    printf("Привет %s", argv[1]);

    return 0;

}

**Указатели**

тип \*имя\_указателя;

int \*pointer;

int &&point2; - указатель на указатель

& - адрес

malloc – попросить у ОС некоторое кол-во памяти

free – для освобождения этой памяти

**18.10.2023/25.10.2023**

**Лекция 7. Лекция 8**

**Указатели. Динамический массив. typedef. struct**

int \*pa

pa = (int \*)malloc(sizeof(int));

free(pa)

int a[80]

int \*p

p = a;

a[3]

p[3]

\*(p+3) – 4 элемент массива

b[0][4]

\*((int\*)b+4)

b[j][k]

\*((базовый тип\*)b+(j\*длину строки)+k)

**Динамический массив**

int \*pa

int n;

printf(“Введите кол-во элементов”);

scanf(“%d”, &n);

pa = (int\*)malloc(sizeof(int)\*n);

pa[5] = 10;

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

scanf(“%d”, (pa+i));

printf(“%d”m \*(pa+i)); - \* чтобы вернуть значение по адресу

}

int \*a[10];

int mas1[50];

a[0] = mas1;

void syntax\_error(int n) {

char \*err[] = {

“Невозможно открыть файл\n”,

“Ошибка при чтении\n”,

“Ошибка при записи\n”,

}

printf(“%s”, err[n]);

}

char \*match(char c, char \*s) {

while (c != \*s && \*s)s++;

return s;

}

#include <stdio.h>

int main() {

char s[80], ch;

gets(s); - считываем строку

ch = getchar(); - считываем символ

ps = match(ch,s);

if(\*p == 0)

printf(“Символ не найден”);

else

printf(“%s”, ps);

return 0;

}

#include <stdio.h>

#include <stdlib.h>

void check(char \*o, char \*b, int (\*cmp)(const \*char, const \*char));

int main() {

    char s1[80], s2[80];

    int (\*p)(const char\*, const char\*);

    p=st2cmp; // ???

    gets(s1);

    gets(s2);

    check(s1, s2, p);

}

void check (char \*a, char\*b, int(\*cmp)(const \*char, const \*char)) {

    printf("Проверка равенства\n");

    if(!(\*cmp)(a,b))

    printf("равны");

    else

    printf("не равны");

}

остальное в vs code

**typedef**

typedef тип новое\_имя

typedef int chislo;

int a;

chislo b; - интовая переменная

**struct**

struct addr {

char city[20];

char street[40];

int zip;

int house;

int apartment;

};

int main() {

struct addr a1;

my-type a2;

a1.zip = 248000;

a2.zip = 248030;

01.11.2023

**Лекция 9.**

struct passports {

char name [80];

int number;

char family\_name[100];

};

typedef struct passports passport;

struct passports person1;

passport person2;

person.number = 100;

printf(“%d”, person.number);

person2 = person1;

int main() {

struct {

int x;

int y;

}

…

struct point a = {5, 11};

func(a.x)

foo(a);

void foo(struct point a);

struct point mas[100];

mas[0].x=100;

mas[0].y=50;

struct person {

char name[80];

int age;

char family.name[100];

};

struct worker {

struct person info;

float salary;

};

struct worker worker1;

worker1.salary = 110,75;

worker.info.age = 40;

worker1.info.name = “Ivan”;

struct person \*person1; - указатель

struct person \*pperson;

struct person person1;

pperson = &person1;

pperson -> age = 45;

Линейные списки:

- стек

- очередь

- деревья

**08.11.2023**

**Лекция 10**

см. vs code

#define – записать макрос

#define LEFT 1

#define DEBUG

#if

#else

#elif

#ifdef DEBUG

#ifndef

#endif

#undef

**Работа с файлами**

FILE \*file;

fopen() – открыть файл для работы с ним

fclose() – закрыть файл

fgets() – читать файл

fputs() –

fseek() – переместить указатель на заданное кол-во байтов

ftell() – получить текущую позицию в байтах относительно начала

fprintf() – печатать в файл

fscanf() – считывать из файла

feof() – возвращает true если достигли конца файла

rewind() – перемотка

remove() – удалить файл

fflush() – очистить поток

**Макросы**

Eof

seek\_set

seek\_cur

seek\_end

**15.11.2023**

**Лекция 11**

r – чтение

w – запись

a – дозапись

rb

wb

ab

r+

…

r+b

w+b

…

FILE \*f;

f = fopen(“text.txt”, “r”);

if (f == NULL)

printf(“Невозможно открыть”);

flose(f);

int fclose(FILE \*fp)

putc

fputc

int putc(int символ, FILE \*fp)

char oh = ‘A’;

putc(ch, f)

fgetc – читает символы с файла

getc

int getc(FILE \*fp);

do {

ch = getc(f);

}

while (ch != EoF);

int feof(FILE \*fp);

while(?feof(fp))ch=getc(fp);

int fflush(FILE \*fp)