Л.р. № 3. Вычисление функций с использованием их разложения в степенной ряд. Программирование алгоритмов циклической структуры.

Цель работы – _____

_____

_____

_____

Домашняя работа:

Сколько оперативной памяти выделит компилятор для хранения данных следующих объявленных переменных?

объявление переменной	объём потребляемой оперативной памяти компьютера (в байтах)
float a;	
short b;	
char c;	
double d;	
long e;	
unsigned char g;	
signed short z;	
long double ld;	

Объявите переменную **n**, которая может принимать при работе программы указанные значения и занимающую минимально возможный объём оперативной памяти.

принимаемые переменной **n** значения	объявление переменной **n**
-2; 106; +213	
-2.7; 10; +1024.3	
0; 1	
‘a’; ‘6’; 4; ‘d’	
60000	
200	

Комментарий это _____

_____

_____

_____

_____

	Комментарий	
	Односторонний	**Двухсторонний**
Возможности:		
Назначение:		
Пример использования:		

Начинающий программист Незнайка написал программу, которая ищет все слова, состоящие из одних и тех же букв.

```
#include <stdio.h>
#define max_length 10000
unsigned char s[max_length];
  int main(){
unsigned char *t;
     while(fgets(s,max_length,stdin)){
t=strchr(s,'\n'); if(t) *t=0;
printf("%d %s ",strlen(s), s);
int i,j,n;unsigned char tmp;n=strlen(s);
  for(i=0;i<n;i++)
for(j=i+1;j<n;j++)if(s[i]>s[j]) tmp=s[i], s[i]=s[j], s[j]=tmp;
printf("%s\n",s);}}
```

ЗАДАНИЕ: написанную Незнайкой программу необходимо переписать с учётом правил оформления программного кода, изложенных в **приложении 9** данной тетради.

ВНИМАНИЕ: в последующем, чтобы проверка преподавателем Ваших программ стала возможной, они должны быть оформлены по вышеуказанным правилам.

Отладка программы

1. Напишите указанную ниже программу.

Откройте Builder, создайте консольное приложение и введите в созданный средой C++ Builder шаблон проекта выделенный текст программы.

```
// Программа "Решение квадратного уравнения"
#include <stdio.h>
#include <math.h>
#include <conio.h>
int main(int argc, char* argv[])
{
    float A,B,C,D,E,X1,X2;
    printf("Input A, B, C:\n");
    scanf("%f%f%f", &A,&B,&C);
    D=pow(B, 2)-4*A*C;
    if (D>=0){
              E=2*A;
              X1=(-B+sqrt(D))/E;
              X2= (-B-sqrt(D))/E;
              printf("X1=%f  X2=%f\n", X1, X2);
    }
    else printf("No result\n");
    getch();
    return 0;
}
```

ПРИМЕЧАНИЕ: Информационная поддержка процесса отладки находится в методическом пособии к лабораторной работе № 3, в разделе "Отладка программы".

2. Изучите диагностические сообщения C++ Builder.

Поочередно внося ошибки в программу, фиксируйте сообщения об ошибках в специальной таблице, представленной ниже. Классифицируйте ошибку, расшифруйте сообщение системы и определите этап выполнения программы (компиляция, компоновка или выполнение), на котором была обнаружена данная ошибка.

Таблица Диагностические сообщения

№	Ошибка	Физический смысл ошибки	Проявление ошибки	Расшифровка сообщения	Этап
1	float AB, C, D, E, X1, X2;	Вместо переменных A и B описана переменная AB	Получено сообщение: Undefined symbol 'A'	Не объявлена переменная A	Компиляция
2	Scanf("%f %f %f", &A, &B, &C);				

3	X1=(-B+ sqrt(D))\E;				
4	D=psw(B,2)-4*A*C;	Вызвана несуществующая функция			
5	D=pow(B)-4*A*C;				
6	Исходные данные: 1 1 3				
7	Исходные данные: 0 1 3				
8	/* *E=2*A;* */ При исходных данных: 1 5 2	Пропущена строка вычислений			
9	scanf("%f%f%f", &A, &C); При исходных данных: 1 5 2	Не определено (не введено) значение B			
10	scanf("%f%f%f", A, B, C); При исходных данных: 1 5 2				

3. Изучите средства отладки программ в среде C++ Builder

Порядок работы:

A. Внесите в программу ошибку 6. Выполните пошаговую трассировку программы, отслеживая значения переменных A, B, C, D, E, X1, X2 в окне Watch. см. раздел “Отладка программы” к лабораторной работе № 3.

Таблица трассировки программы

Команда	Значения переменных и условий							
	A	***B***	***C***	***D***	***E***	***X1***	***X2***	***D>=0***
printf("Input A, B, C:\n");								
scanf("%f %f %f", &A, &B, &C);								
D=pow(B, 2)-4*A*C;								
if (D>=0){								
E=2*A;								
X1=(-B+sqrt(D))/E;								
X2= (-B-sqrt(D))/E;								
printf("X1=%f X2=%f\n", X1, X2); }								
else printf("No result\n");								
getch();								
return 0;								

ПРИМЕЧАНИЕ: Вы должны запустить программу в отладочном (пошаговом) режиме. Проходя по каждой строчке программы (команде), записывая в таблицу значения переменных и условий.

Какое значение имеет переменная **D**? В какой момент обнаруживается ошибка?

_____

_____

_____

_____

Б. Внесите в программу ошибку 7. Выполните пошаговую трассировку программы, отслеживая значение переменных A, B, C, D, E, X1, X2.

Таблица трассировки программы

Команда	Значения переменных и условий							
	A	***B***	***C***	***D***	***E***	***X1***	***X2***	***D>=0***
printf("Input A, B, C:\n");								
scanf("%f %f %f", &A, &B, &C);								
D=pow(B, 2)-4*A*C;								
if (D>=0){								
E=2*A;								
X1=(-B+sqrt(D))/E;								
X2= (-B-sqrt(D))/E;								
printf("X1=%f X2=%f\n", X1, X2); }								
else printf("No result\n");								
getch();								
return 0;								

Какое значение имеет переменная **E** в момент вычисления корней уравнения? Почему?

_____

_____

_____

_____

В. Установите точку останова перед вычислением дискриминанта. Выполните программу до точки останова. Просмотрите значения переменных, подводя к ним курсор мыши.

Переменная	***A***	***B***	***C***	***D***	***E***	***X1***	***X2***
Значение							

4. Используя отладочный режим определите предназначение функций

```
int F1(){
     char Action[] = {85, -117, -20, -72, 5, 0, 0, 0, 93, -61};
     return ((int (*)()) Action)();
}
```

выполняет _____

```
int F2(int x, int y) {
     char Action[] = {85, -117, -20, -117, 69, 8, -9, 109, 12, 93, -61};
     return ((int (*)(int, int)) Action)(x,y);
}
```

выполняет _____

```
int F3(int x, int y) {
     char Action[] = {85, -117, -20, -117, 69, 8, 43, 69, 12, 93, -61};
     return ((int (*)(int, int)) Action)(x,y);
}
```

выполняет _____

```
int F4(int x, int y) {
   char Action[]={85,-117,-20,-117,69,8,-103,-9,125,12,-117,-62,93,-61};
   return ((int (*)(int, int)) Action)(x,y);
}
```

выполняет _____

```
int F5(int x, int y) {
     char Action[] = {85, -117, -20, -117, 69, 8, 3, 69, 12, 93, -61};
     return ((int (*)(int, int)) Action)(x,y);
}
```

выполняет _____

```
int F6(int x, int y) {
     char Action[]={85,-117,-20,-117,69,8,-103,-9,125,12,93,-61};
     return ((int (*)(int, int)) Action)(x,y);
}
```

выполняет _____

```
int F7(int x, int y) {
     char Action[]={85,-117,-20,-117,69,8,-125,-64,2,-9,109,12,93,-61};
     return ((int (*)(int, int)) Action)(x,y);
}
```

выполняет _____

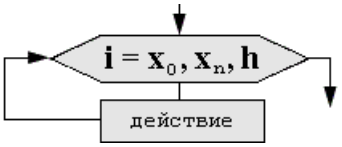
```
int F8(int x, int y) {
     char Action[]={85,-117,-20,-117,69,8,-71,3,0,0,0,-103,-9,-7,-117,
                                                  -62,-9,109,12,93,-61};
     return ((int (*)(int, int)) Action)(x,y);
}
```

выполняет _____

Краткая теория

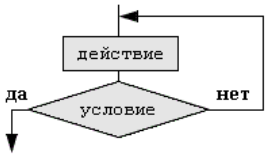
Блок-схема алгоритма вычисления суммы n членов степенного ряда

ВНИМАНИЕ: в алгоритме вычисления суммы **n** членов степенного ряда необходимо использовать цикл с параметром.



Блок-схема алгоритма вычисления суммы ряда с учётом заданной точности ε

ВНИМАНИЕ: в алгоритме вычисления суммы ряда с учётом заданной точности необходимо использовать цикл с постусловием.



Содержание краткой теории: Зачем функцию раскладывать в ряд? Что такое цикл? Какие бывают виды циклов? Что такое арифметический цикл? Что такое итерационный цикл? Чем отличается арифметический цикл от итерационного цикла? Блок-схема алгоритма вычисления суммы ряда для заданного количества членов. Блок-схема алгоритма вычисления суммы степенного ряда с учётом заданной точности.

Что такое венгерская нотация? Зачем нужна венгерская нотация? Примеры объявления переменных и функций с учётом венгерской нотации.

Постановка задачи: для **x** изменяющегося от **a** до **b** с шагом **(b-a)/10** вычислить функцию **f(x)**:

а) используя ее разложение в степенной ряд для заданного количества членов ряда **n**;

б) используя ее разложение в степенной ряд для заданной точности **ε**;

в) используя стандартные библиотечные функции.

Задание:

(функция, степенной ряд, диапазон изменения аргумента, n, ε)

Блок - схема алгоритма

Текст программы (листинг)

ВНИМАНИЕ: Программа должна быть написана с использованием ***венгерской нотации*** и по всем правилам оформления программного кода.

ch
c
sz
str
s
n
i
u
l
ul
ff
dbl
kdbl

Результаты работы программы

X=_____	SN=_____	SE=_____	Y=_____
X=_____	SN=_____	SE=_____	Y=_____
X=_____	SN=_____	SE=_____	Y=_____
X=_____	SN=_____	SE=_____	Y=_____
X=_____	SN=_____	SE=_____	Y=_____
X=_____	SN=_____	SE=_____	Y=_____
X=_____	SN=_____	SE=_____	Y=_____
X=_____	SN=_____	SE=_____	Y=_____
X=_____	SN=_____	SE=_____	Y=_____
X=_____	SN=_____	SE=_____	Y=_____
X=_____	SN=_____	SE=_____	Y=_____
X=_____	SN=_____	SE=_____	Y=_____

ПРИМЕЧАНИЕ: **X**- значение входного параметра для рассчитываемой функции (**X** меняется от **a** до **b** с шагом **(b-a)/10**); **SN**- значение суммы степенного ряда для заданного количества членов **n**; **SE**- значение суммы степенного ряда для заданной точности **ε**; **Y**- значение функции, вычисленное с использованием стандартных библиотечных функций.

Какая точность (ε) расчёта значения Y получилась у SN? _____

© Г.В. Мальшаков, 2016