Пример вычисления значения многочлена спомощью схемы Горнера и с помощью задания функции пользователя.

1. Схема Горнера. p(x) = x4 – 6x3 + 11 x2 - 6x-15

a0 a1  a2 . . . an | x

+ b0\*x b1\*x. . . bn-1\*x

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

b0 b1 b2 . . . bn == P(x)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2.Вычисление значения функции с помощью создания функции пользователя.

#include <stdio.h>

#include <stdlib.h>

int poly(int a[],int n,int x);

int gorner(int a[],int b[], int n,int x);

int main()

{

int s,t,x;

const int n=5;

int a[n]={1,-6,11,-6,-15}, b[n];

x=4;

s=gorner(a,b,n,x);

t =poly(a,n,x);

printf("s=%i p= %i \n",s,t);

system("pause");

return 0;

}

Функция пользователя

int poly(int a[],int n,int x)

{

int r=1,s=a[n-1];

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

{r\*=x;

s+=a[n-i-2]\*r;

}

return(s);

}

Схема Горнера

int gorner(int a[],int b[], int n,int x)

{

b[0]=a[0];

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

b[i]=a[i]+b[i-1]\*x;

return (b[n-1]);

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. Вычисление значений функций заданных рядом.

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

float expon(float x);

int main()

{

float x, s,r;

x=1;

s=expon(x);

r=exp(x);

printf("S=%f exp = %f\n",s,r);

system("pause");

return 0;

}

float expon(float x)

{

float s=1,a=1,e=0.001;

int k=0;

do{

k++;

a\*=x/k;

s+=a;

}

while(a>e);

return (s);

} э

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4 .Вычисление кусочнозаданных функций.



//Вычисление значений кусочно- заданной функции.

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int main()

{

float kus(float xn, float xk,float d,float y[],int m);

float xn,xk,d,r;

int n,m;

m=10;

float y[m];

printf("input znathenie Xn Xk D \n");

scanf("%f%f%f",&xn,&xk,&d);

n=(int)(xk-xn)/d;

r=kus(xn,xk,d,y,n);

system("pause");

return 0;

}

float kus(float xn, float xk,float d,float y[],int m)

{

const float a=5,b=7,c=3;

float f;

f=0;

for (float x=xn;x<=xk;x+=d)

{

if(x!=0) f=-x/c-c/2;

if(x<0&&b!=0&&(c\*x-a)!=0) f=-(2\*x)/(c\*x-a);

if(x>0&&b==0&&(x-c)!=0) f=(x-a)/(x-c);

printf("x= %f f= %f\n",x,f);

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5. Итерационные алгоритмы.

1.Вычисления квадратного корня.

#include <stdio.h>

#include <stdlib.h>

#include<math.h>

float sqr2(float x);

int main()

{

float x,r;

x=4;

r=sqr2(x);

printf("|/x=%f\n",r);

system("pause");

return 0;

}

float sqr2(float x)

{

float y,y1,e1,e=0.0001;

y=x;

do{

y1=(y+x/y)/2.;

e1=fabs(y-y1);

y=y1;

}

while(e1>e);

return (y);

}

2. Вычисление обратных величин.

#include <stdio.h>

#include <stdlib.h>

#include<math.h>

float obrat(float x);

int main()

{

float x,r;

x=4;

r=obrat(x);

printf("|/x=%f\n",r);

system("pause");

return 0;

}

float obrat(float x)

{

float y,y1,e1,e=0.0001;

y=x/10.;

do{

y1=y\*(2.-x\*y);

e1=fabs(y-y1);

y=y1;

}

while(e1>e);

return (y);

}

 #include <stdio.h>

#include <stdlib.h>

#include<math.h>

float obrat(float x);

int main()

{

float x,r;

x=4;

r=obrat(x);

printf("|/x=%f\n",r);

system("pause");

return 0;

}

float obrat(float x)

{

float y,y1,e1,e=0.0001;

y=x/10.;

do{

y1=y\*(3.-x\*y\*y)/2.;

e1=fabs(y-y1);

y=y1;

}

while(e1>e);

return (y);

}

  

Вычисление корня кубического

#include <stdio.h>

#include <stdlib.h>

#include<math.h>

float root3(float x);

int main()

{

float x,r;

x=27;

r=root3(x);

printf("|3/=%f\n",r);

system("pause");

return 0;

}

float root3(float x)

{

float y,y1,e1,e=0.0001;

y=x;

do{

y1=(2.\*y+x/y/y)/3.;

e1=fabs(y-y1);

y=y1;

printf("%f\n",y);

}

while(e1>e);

return (y);

}