**Комп‘ютерний практикум №4**

**Тема:** Оператори циклу. Робота з масивами.

**Завдання 1:** Написати програму, яка повинна виводити таблицю значень синусів або косинусів (розрахованих за допомогою розкладання функції в ряд Тейлора) і табличних значень, а також їх різницю в заданому діапазоні із заданим кроком та точністю.

***Текст програми:***

#include <stdlib.h>

#include <malloc.h>

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

code()

{

char c;

int n;

int i, j;

int q, k;

int eo;

float \*\*a, \*b, \*E, \*xp, \*Ech, \*x, \*d;

float eps, prop, m;

do

{

printf("Enter number of variable: ");

if ( (scanf("%d%c", &n, &c) != 2) || (c != '\n') || (n <= 1) || (n > 3e100) )

{

printf("Enter only natural numbers from 2 to 3e100\n");

fflush(stdin);

j = 0;

} else

{

j = 1;

}

} while(j != 1);

do {

printf("\nEnter accuracy (<=6): ");

if ( (scanf("%d%c", &eo, &c) != 2) || (c != '\n') || ((eo != 0) && (eo != 1) && (eo != 2) && (eo != 3) && (eo != 4) && (eo != 5) && (eo != 6)) )

{

printf("Enter only integer numbers from 0 to 6\n");

fflush(stdin);

j = 0;

} else

{

j = 1;

}

} while(j != 1);

/\*switch (eo) {

case 0: eps = 1; break;

case 1: eps = 0.1; break;

case 2: eps = 0.01; break;

case 3: eps = 0.001; break;

case 4: eps = 0.0001; break;

case 5: eps = 0.00001; break;

case 6: eps = 0.000001; break;

}

\*/

{

eps=1/pow(10,eo);

}

a = (float\*\*) calloc(n,sizeof(float\*));

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

{

a[i] = (float\*)calloc(n, sizeof(float\*));

}

b = (float\*)calloc(n, sizeof(float));

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

{

for (j=0; j<n; j++)

{

printf ("a[%d][%d]=", i+1, j+1);

do

{

if ( (scanf ("%f%c", &a[i][j], &c)!=2)||(a[i][j]>3e100)||(a[i][j]<-3e100)||((a[i][j]==0)&&(i==j))||(c!='\n') )

{

printf ("Be careful, enter only numbers from -3k to 3k & the main diagonal != 0\n");

printf ("a[%d][%d]=", i+1, j+1);

fflush(stdin);

k=0;

}

else {k=1;}

}

while(k==0);

}

do

{

printf ("b[%d]=", i+1);

if ( (scanf ("%f%c", &b[i], &c)!=2)||(b[i]>3e10)||(b[i]<-3e10) )

{

printf ("Be careful, enter only numbers from -3k to 3k\n");

fflush(stdin);

k=0;

}

else {k=1;}

}

while(k==0);

}

printf ("Your sysiem is:\n");

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

{

for(j=0; j<=n; j++)

{

if (j==n)

{

printf("=%g", b[i]);

}

else

printf("+(%g\*x%d) ", a[i][j], j+1);

}

printf("\n");

}

printf ("Tap Enter to continue\n");

getch();

Ech = (float\*)calloc(n, sizeof(float));

//proverka na symmy

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

{

Ech[i]=0;

for (j=0; j<n; j++)

{

if (i!=j)

{

Ech[i] += a[i][j];

}

}

if (fabs(a[i][i])<fabs(Ech[i]))

{

printf("Error, this SLAE have no solutions ( abs(a[i][i]) < abs(sum(a[i][j])) )\n");

return 0;

}

}

xp=(float\*)calloc(n,sizeof(float));

//podschet xp

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

{

xp[i]=b[i]/a[i][i];

}

E=(float\*)calloc(n,sizeof(float));

x=(float\*)calloc(n,sizeof(float));

d = (float\*)calloc(n, sizeof(float));

//podschet x

do

{

int p;

prop=0;

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

{

E[i] = 0;

for (j=0; j<n; j++)

{

if (i!=j)

{

E[i]=E[i]+(a[i][j]\*xp[j]);

}

}

x[i]=(b[i]-E[i])/a[i][i];

d[i]=fabs(x[i]-xp[i]);

if (d[i]>prop)

{

prop=d[i];

}

xp[i]=x[i];

p++;

if (p==100)

{

printf("Error, this SLAE have no solutions ( cycle=100 )\n");

return 0;

}

}

} while(prop >= eps);

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

{

printf("x%d=%f\n", i+1, xp[i]);

}

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

{

free(a[i]);

}

free(a);

free(b);

free(Ech);

free(xp);

free(E);

free(x);

free(d);

}

main()

{

char exitchar = 'y';

do

{

system("cls");

code();

printf ("Enter y to continue\n");

exitchar = getch();

fflush(stdin);

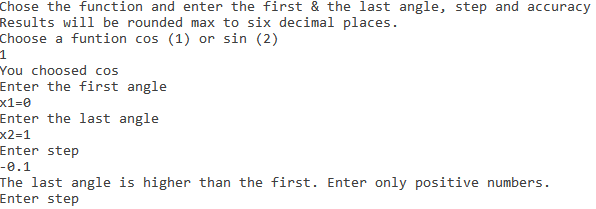
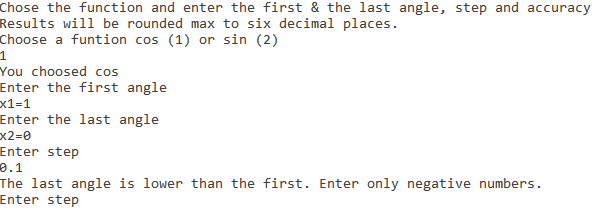
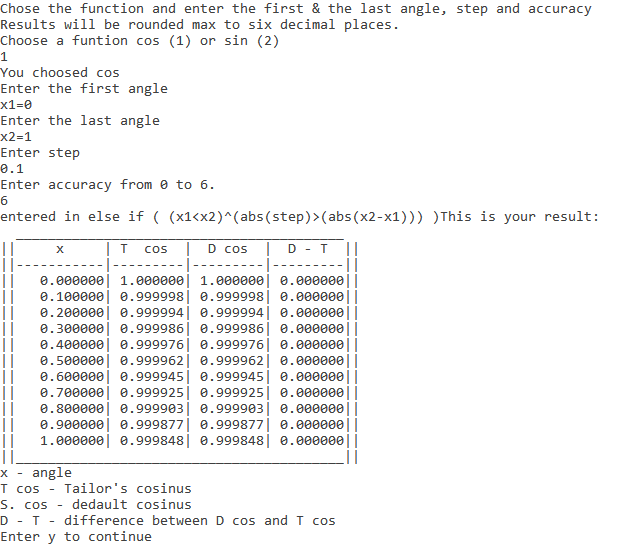
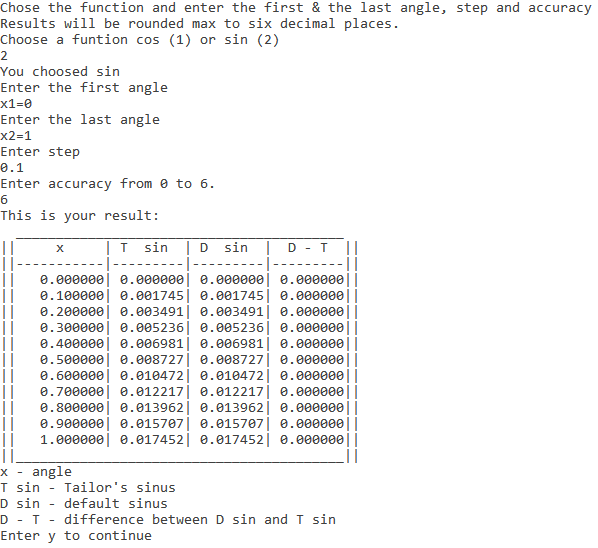
}

while ( (exitchar == 'y')||(exitchar == 'Y') );

return 0;

}

***Схема до програми:***



***Висновки****: Програма вирішує поставлене завдання. Теоретичні розрахунки відповідають отриманим. Програма працює корректно.*