**Задание № 4**

**Индивидуальное задание:**

Написать программу формирования ОПЗ и расчета полученного выражения. Разработать удобный интерфейс ввода исходных данных и вывода результатов. Работу программы проверить на конкретном примере.





* 1. **Создание консольного приложения**

Код программы:

#include <stdio.h>

#include <string.h>

#include <Windows.h>

double GetValue(char Sym);

double CalculateExpression(char\* Str);

/\*

Exp = (a+b)\*(c-d)/e

\*/

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

char vihod[100];

char stek[100];

char a[100];

int j = 0, m = 0, i = 0;

int len;

printf("%s", "Введите строку\n");

gets(a);

len = strlen(a);

while ((i < len) && (a[i] != '0' || a[i] != '='))

{

if (a[i] >= 'a' && a[i] <= 'z')

{

vihod[m] = a[i];

m++;

}

else if (a[i] == '(')

{

stek[j] = a[i];

j++;

}

else if (a[i] == '+' || a[i] == '-' || a[i] == '/' || a[i] == '\*')

{

if (j == 0)

{

stek[j] = a[i];

j++;

}

else

{

if (prioritet(stek[j - 1]) < prioritet(a[i]))

{

stek[j] = a[i];

j++;

}

else

{

while (j >= 1 && prioritet(stek[j - 1]) >= prioritet(a[i]))

{

vihod[m] = stek[j - 1];

m++;

j--;

}

stek[j] = a[i];

++j;

}

}

}

if (a[i] == ')')

{

while (stek[j - 1] != '(')

{

vihod[m] = stek[j - 1];

j--;

m++;

}

j--;

}

i++;

}

--j;

while (j >= 0)

{

vihod[m] = stek[j];

++m;

--j;

}

vihod[m] = '\0';

printf("%lf\n", CalculateExpression(vihod));

system("pause");

return 0;

}

int prioritet(char a)

{

switch (a)

{

case '\*':

case '/':

return 3;

case '-':

case '+':

return 2;

case '(':

return 1;

}

}

double GetValue(char Sym) {

switch (Sym)

{

case 'a': return 0.8;

case 'b': return 4.1;

case 'c': return 7.9;

case 'd': return 6.2;

case 'e': return 3.5;

}

}

double CalculateExpression(char\* Str) {

char CurSym;

double tmpOpnd1, tmpOpnd2, Result, Stack[100];

int j = 0;

int i = 0;

while (i < strlen(Str))

{

CurSym = Str[i++];

if (CurSym >= 'a' && CurSym <= 'z')

{

Stack[j++] = GetValue(CurSym);

}

else if (CurSym >= '0' && CurSym <= '9')

{

Stack[j++] = GetValue(CurSym - 0x30);

}

else

{

tmpOpnd1 = Stack[--j]; tmpOpnd2 = Stack[--j];

switch (CurSym) {

case '+':

Result = tmpOpnd2 + tmpOpnd1;

break;

case '-':

Result = tmpOpnd2 - tmpOpnd1;

break;

case '\*':

Result = tmpOpnd2 \* tmpOpnd1;

break;

case '/':

Result = tmpOpnd2 / tmpOpnd1;

break;

case '^':

Result = pow(tmpOpnd2, tmpOpnd1);

break;

}

Stack[j++] = Result;

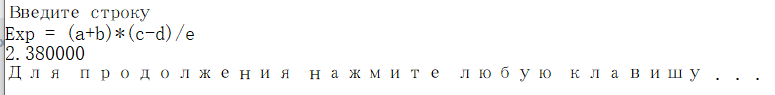
}

}

return Stack[--j];

}

**Результаты выполнения:**

****