#include <iostream>

#include "STACKPAC.h"

int main()

{

STACK <int, 10> S;

S.clearStack();

int a = 5, b = 7, c = 8, d=2;

int x, y, r;

char p;

char again = 'y';

do

{

std::cout << "Enter a postfix expression : ";

std::cin >> p;

while (p != '$')//

{

switch (p)

{

case('a'):

S.pushStack(a);

break;

case('b'):

S.pushStack(b);

break;

case('c'):

S.pushStack(c);

break;

case('d'):

S.pushStack(d);

break;

case('+'):

x = S.popStack();

y = S.popStack();

r = y + x;

S.pushStack(r);

break;

case('-'):

x = S.popStack();

y = S.popStack();

r = y - x;

S.pushStack(r);

break;

case('\*'):

x = S.popStack();

y = S.popStack();

r = y \* x;

S.pushStack(r);

break;

case('/'):

x = S.popStack();

y = S.popStack();

r = y / x;

S.pushStack(r);

break;

}//end switch

std::cin >> p;

}//endwhile

using std::endl;

std::cout << "The value of the expressions is: " << S.popStack() << endl;

std::cout << "CONTINUE(y/n)? ";

std::cin >> again;

} while ((again == 'y') || (again == 'Y'));

std::system("pause");

return 0;

}

//################################

HEADER FILE

#pragma once

#ifndef STACKPAC

#define STACKPAC

template<class T, int n>

class STACK

{

private:

T a[n];//T for typ, n for length

int counter;

public:

void clearStack()

{

counter = 0;

}//end clearStack

bool emptyStack()

{

return (counter == 0) ? true : false;

}//end emptyStack

bool fullStack()

{

return (counter == n) ? true : false;

}//end fullStack

void pushStack(T x)

{

a[counter] = x;

counter++;

}//end pushStack

T popStack()

{

counter--;

return a[counter];

}//end popStack

};//end class STACK

#endif