Evaluating postfix

Program:-

#include <stdio.h>

#include <ctype.h>

#define MAXSTACK 100 /\* for max size of stack \*/

#define POSTFIXSIZE 100 /\* define max number of charcters in postfix expression \*/

/\* declare stack and its top pointer to be used during postfix expression

evaluation\*/

int stack[MAXSTACK];

int top = -1; /\* because array index in C begins at 0 \*/

/\* can be do this initialization somewhere else \*/

/\* define push operation \*/

void push(int item)

{

if (top >= MAXSTACK - 1) {

printf("stack over flow");

return;

}

else {

top = top + 1;

stack[top] = item;

}

}

/\* define pop operation \*/

int pop()

{

int item;

if (top < 0) {

printf("stack under flow");

}

else {

item = stack[top];

top = top - 1;

return item;

}

}

/\* define function that is used to input postfix expression and to evaluate it \*/

void EvalPostfix(char postfix[])

{

int i;

char ch;

int val;

int A, B;

/\* evaluate postfix expression \*/

for (i = 0; postfix[i] != ')'; i++) {

ch = postfix[i];

if (isdigit(ch)) {

/\* we saw an operand,push the digit onto stack

ch - '0' is used for getting digit rather than ASCII code of digit \*/

push(ch - '0');

}

else if (ch == '+' || ch == '-' || ch == '\*' || ch == '/') {

/\* we saw an operator

\* pop top element A and next-to-top elemnet B

\* from stack and compute B operator A

\*/

A = pop();

B = pop();

switch (ch) /\* ch is an operator \*/

{

case '\*':

val = B \* A;

break;

case '/':

val = B / A;

break;

case '+':

val = B + A;

break;

case '-':

val = B - A;

break;

}

/\* push the value obtained above onto the stack \*/

push(val);

}

}

printf(" \n Result of expression evaluation : %d \n", pop());

}

int main()

{

int i;

/\* declare character array to store postfix expression \*/

char postfix[POSTFIXSIZE];

printf("ASSUMPTION: There are only four operators(\*, /, +, -) in an expression and operand is single digit only.\n");

printf(" \nEnter postfix expression,\npress right parenthesis ')' for end expression : ");

/\* take input of postfix expression from user \*/

for (i = 0; i <= POSTFIXSIZE - 1; i++) {

scanf("%c", &postfix[i]);

if (postfix[i] == ')') /\* is there any way to eliminate this if \*/

{

break;

} /\* and break statement \*/

}

/\* call function to evaluate postfix expression \*/

EvalPostfix(postfix);

return 0;

}

Algorithm:-

Step 1:Create a stack to store operands.

Step 2:Scan the given expression from left to right.

Step 3:If the scanned character is an operand push it,if its an operator pop 2 operands from stack.

Step 4:Repeat this until all the characters are scanned.

Step 5:When the expression is ended,the number in the stack is the final result.

