**Q. WAP to convert a given infix expression to postfix form using stacks.**

#include <ctype.h>

#include <string.h>

#define size 100

char stack[size];

char infix[size], postfix[size];

int top = -1, choice;

void push(char n)

{

if (top >= size - 1)

{

printf("\nOVERFLOW");

}

else

{

top = top + 1;

stack[top] = n;

}

}

char pop()

{

char n;

if (top < 0)

{

printf("UNDERFLOW : Invalid Infix Expression");

getchar();

exit(1);

}

else

{

n = stack[top];

top = top - 1;

return (n);

}

}

int is\_operator(char symbol)

{

if (symbol == '^' || symbol == '\*' || symbol == '/' || symbol == '+' || symbol == '-')

{

return 1;

}

else

{

return 0;

}

}

int precedence(char symbol)

{

if (symbol == '^')

{

return (3);

}

else if (symbol == '\*' || symbol == '/')

{

return (2);

}

else if (symbol == '+' || symbol == '-')

{

return (1);

}

else

{

return (0);

}

}

void infix\_to\_postfix(char infix\_exp[], char postfix\_exp[])

{

int i, j;

char n, x;

push('(');

strcat(infix\_exp, ")");

i = 0, j = 0;

n = infix\_exp[i];

while (n != '\0')

{

if (n == '(')

{

push(n);

}

else if (isdigit(n) || isalpha(n))

{

postfix\_exp[j] = n;

j++;

}

else if (is\_operator(n) == 1)

{

x = pop();

while (is\_operator(x) == 1 && precedence(x) >= precedence(n))

{

postfix\_exp[j] = x;

j++;

x = pop();

}

push(x);

push(n);

}

else if (n == ')')

{

x = pop();

while (x != '(')

{

postfix\_exp[j] = x;

j++;

x = pop();

}

}

else

{

printf("\n Invalid Infix Expression \n");

getchar();

exit(1);

}

i++;

n = infix\_exp[i];

}

if (top > 0)

{

printf("\n Invalid Infix Expression \n");

getchar();

exit(1);

}

postfix\_exp[j] = '\0';

}

int main()

{

do {

printf("\n Enter Infix Expression : ");

scanf("%s", infix);

infix\_to\_postfix(infix, postfix);

printf("Equivalent Postfix Expression : %s ", postfix);

printf("\nDo you want to convert another expression ?(Yes-1,No-0) : ");

scanf("%u", &choice);

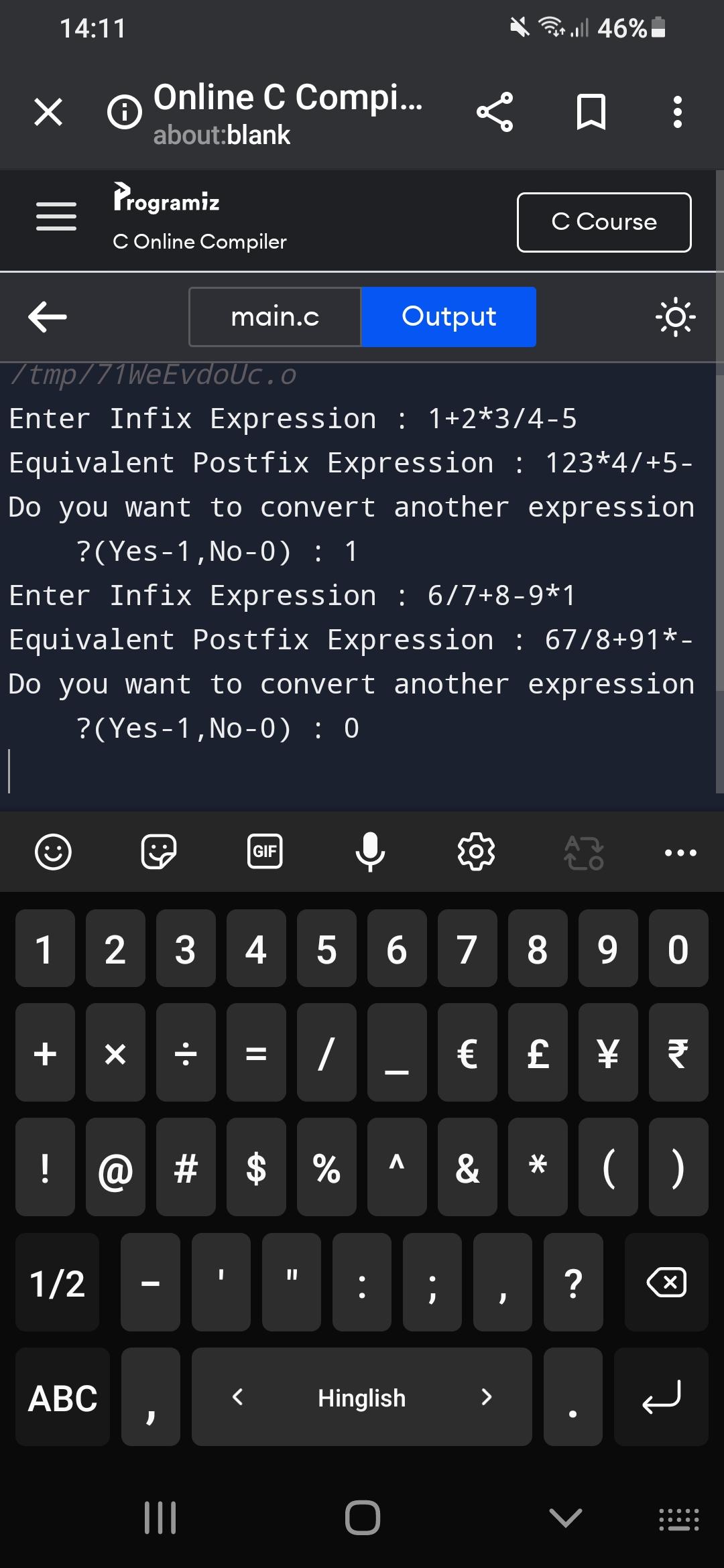
}

while (choice != 0);

return (0);

}

**OUTPUT :**

****