1. To implement Selection and Insertion Sort recursively.
2. Implementation of Selection Sort Recursively

#include<stdio.h>

void selectionsort(int ar[], int i,int size)

{

if (i>=size-1)

return;

int min=i;

for (int j=i+1;j<size;j++ )

{

if (ar[j]<ar[min])

min=j;

}

int temp = ar[i];

ar[i] = ar[min];

ar[min]=temp;

selectionsort(ar,i+1,size);

}

main()

{

int ar[20],n;

printf("Enter the total number of elements to be sorted ");

scanf("%d",&n);

printf("\nEnter the elements for sorting\n");

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

scanf("%d",&ar[i]);

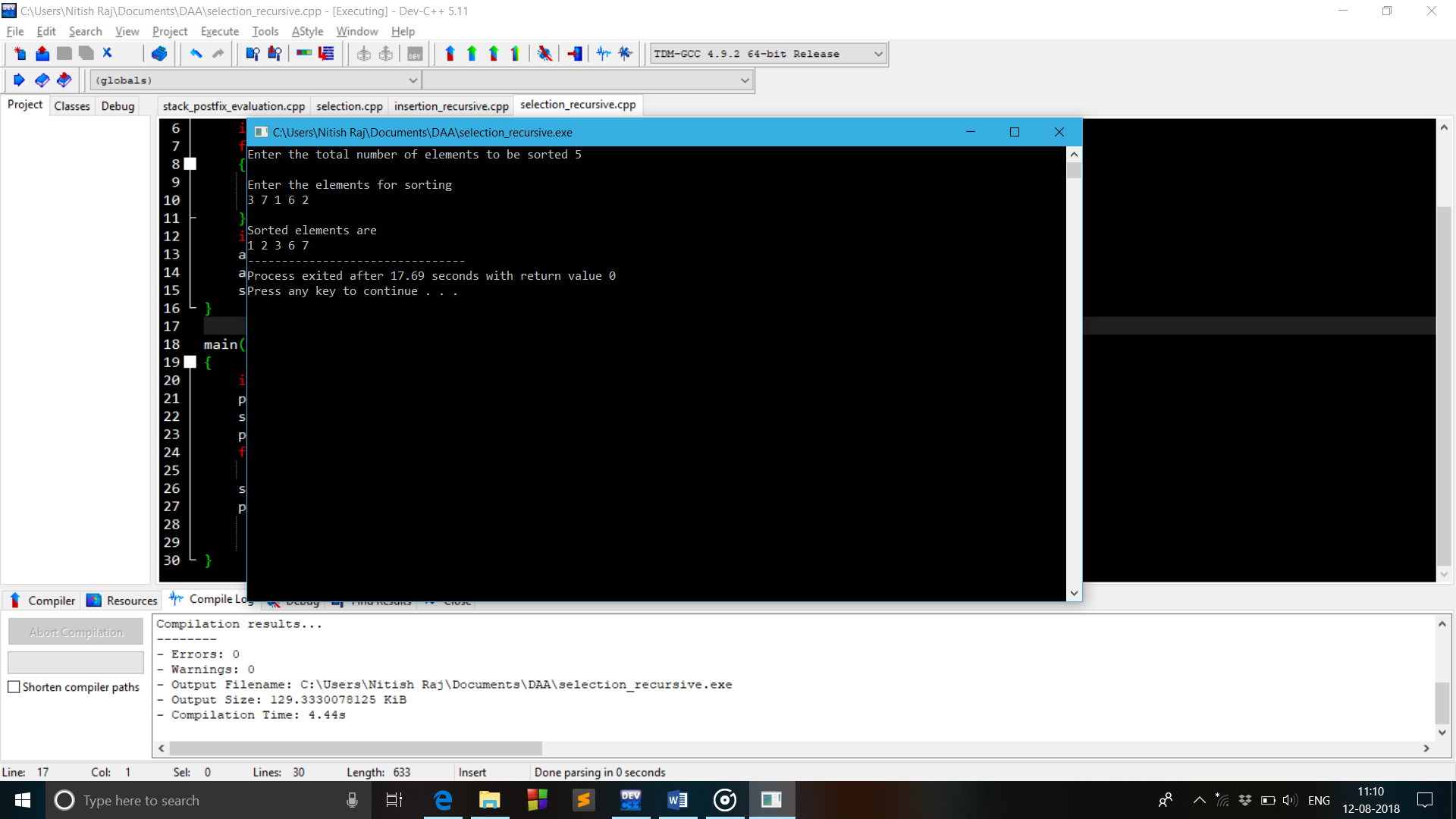
selectionsort(ar,0,n);

printf("\nSorted elements are\n");

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

printf("%d ",ar[i]);

}



1. Implementation of Insertion Sort Recursively

#include<stdio.h>

void insertionsort(int ar[], int i,int size)

{

if(i>size)

return;

int beg=ar[i];

int j=i-1;

while((j>=0)&&(ar[j]>beg))

{

ar[j+1]=ar[j];

j--;

}

ar[j+1]=beg;

insertionsort(ar,i+1,size);

}

main()

{

int n;

printf("Enter the total number of elements to be sorted ");

scanf("%d",&n);

int ar[n];

printf("\nEnter the elements for sorting\n");

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

scanf("%d",&ar[i]);

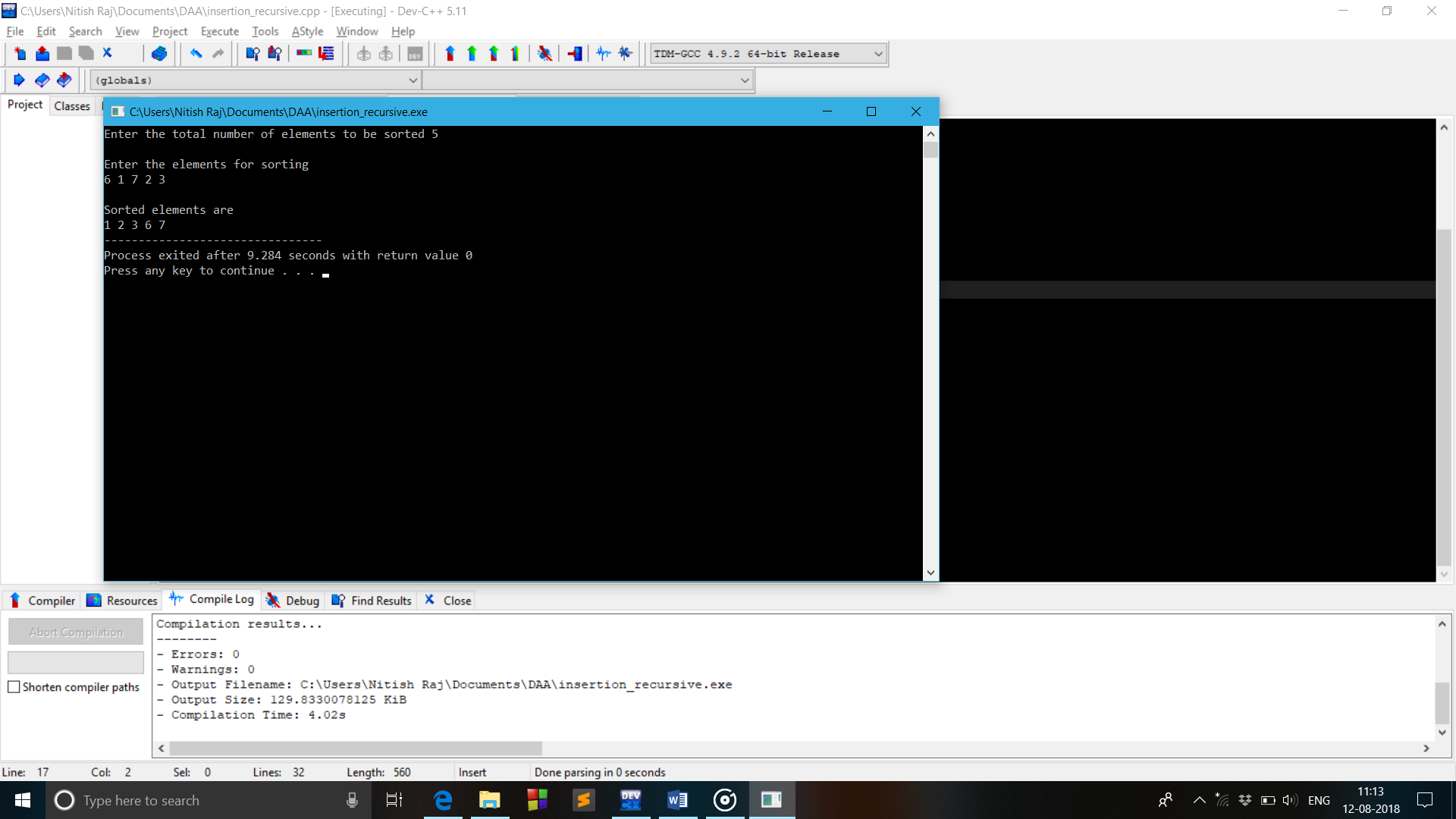
insertionsort(ar,1,n);

printf("\nSorted elements are\n");

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

printf("%d ",ar[i]);

}



1. Conversion of Infix to Postfix expression

#include<stdio.h>

#include<ctype.h>

char stack[20];

int top = -1;

void push(char x)

{

stack[++top] = x;

}

char pop()

{

if(top == -1)

return -1;

else

return stack[top--];

}

int priority(char x)

{

if(x == '(')

return 0;

if(x == '+' || x == '-')

return 1;

if(x == '\*' || x == '/')

return 2;

}

void conversion(char e[])

{

printf(“\nExpression after converting from Infix to Postfix is \n”);

char x;

while(\*e != '\0')

{

if(isalnum(\*e))

printf("%c",\*e);

else if(\*e == '(')

push(\*e);

else if(\*e == ')')

{

while((x = pop()) != '(')

printf("%c", x);

}

else

{

while(priority(stack[top]) >= priority(\*e))

printf("%c",pop());

push(\*e);

}

e++;

}

while(top != -1)

{

printf("%c",pop());

}

}

main()

{

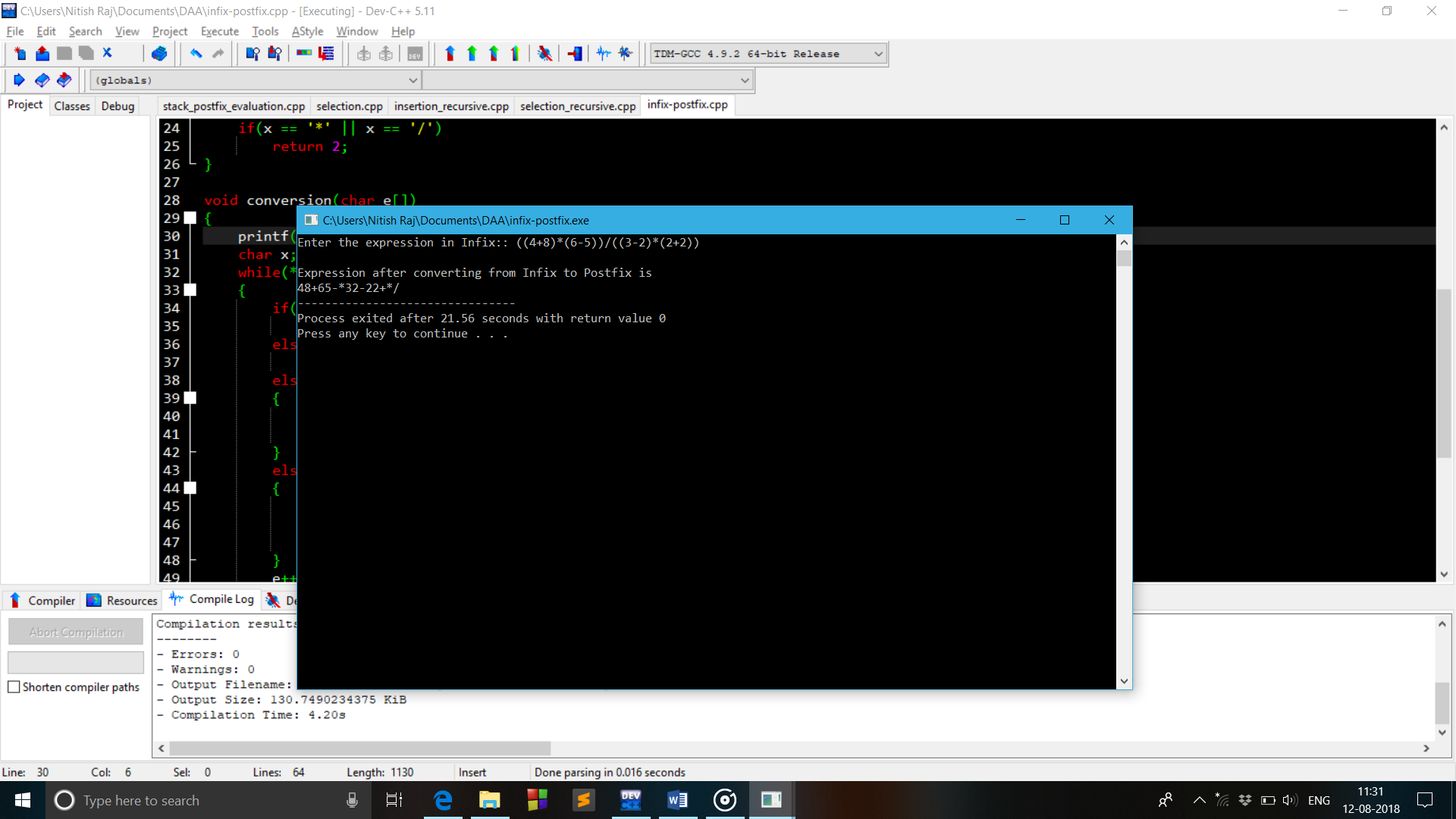
char exp[20];

printf("Enter the expression in Infix :: ");

scanf("%s",exp);

conversion(exp);

}



1. Evaluation of Postfix Expression

#include<stdio.h>

#include<ctype.h>

#include<string.h>

const int size=20;

int stack[size];

int top=-1;

int push(int x)

{

stack[++top]=x;

return 0;

}

int pop()

{

return(stack[top--]);

}

int postfixeval(char ar[],int size)

{

for(int i=0;i<size;i++)

{

if(isdigit(ar[i]))

push(ar[i]-'0');

else

{

int n1=pop();

int n2=pop();

switch(ar[i])

{

case '+':push(n1+n2);

break;

case '-':push(n2-n1);

break;

case '\*':push(n2\*n1);

break;

case '/':push(n2/n1);

break;

default : printf("\nError!!");

break;

}

}

}

return stack[top];

}

int main()

{

char ar[size];

printf("Enter the postfix ");

scanf("%s",ar);

int l=strlen(ar);

float ans = postfixeval(ar,l);

printf("\n\nThe evaluation of postfix expression %s is %f",ar,ans);

}

