CSE 2003

DATA STRUCTURES AND ALGORITHMS

Lab Assessment – 2

L19+L20 | SJT317

FALL SEMESTER 2020-21

by

SHARADINDU ADHIKARI

19BCE2105

Question 1

Problem:

Write a program for converting the given decimal number into binary.

Code:

```
#include <stdio.h>
#include <math.h>

long decimalToBinary(int decimalnum)
{
    long binarynum = 0;
    int rem, temp = 1;

    while (decimalnum!=0)
    {
        rem = decimalnum%2;
        decimalnum = decimalnum / 2;
        binarynum = binarynum + rem*temp;
        temp = temp * 10;
    }
    return binarynum;
}

int main()
{
    int decimalnum;
    printf("Enter a Decimal Number: ");
    scanf("%d", &decimalnum);
    printf("Equivalent Binary Number is: %ld",
decimalToBinary(decimalnum));
    return 0;
}
```

Input:

```
Enter a Decimal Number: 210
```

Output:

```
Equivalent Binary Number is: 11010010
```

Screenshots of Code & CMD:

main.c [LA 2] - Code::Blocks 20.03

```c
#include <stdio.h>
#include <math.h>

long decimalToBinary(int decimalnum)
{
    long binarynum = 0;
    int rem, temp = 1;

    while (decimalnum!=0)
    {
        rem = decimalnum%2;
        decimalnum = decimalnum / 2;
        binarynum = binarynum + rem*temp;
        temp = temp * 10;
    }
    return binarynum;
}

int main()
{
    int decimalnum;
    printf("Enter a Decimal Number: ");
    scanf("%d", &decimalnum);
    printf("Equivalent Binary Number is: %ld", decimalToBinary(decimalnum));
    return 0;
}
```

```
---------- Build: Debug in LA 2 (compiler: GNU GCC Compiler)----------
gcc.exe -Wall -g  -c "C:\Users\Sharadindu\Desktop\DSA Lab Assignments\Q1\main.c" -o obj\Debug\main.o
gcc.exe  -o "bin\Debug\LA 2.exe" obj\Debug\main.o
Output file is bin\Debug\LA 2.exe with size 54.17 KB
Process terminated with status 0 (0 minute(s), 1 second(s))
0 error(s), 0 warning(s) (0 minute(s), 1 second(s))
```

```
"C:\Users\Sharadindu\Desktop\DSA Lab Assignments\Q1\bin\Debug\LA 2.exe"
Enter a Decimal Number: 210
Equivalent Binary Number is: 11010010
Process returned 0 (0x0)   execution time : 1.915 s
Press any key to continue.
```

Question 2

Problem:

Write a program to implement a stack in an array and perform PUSH, POP, PEEP and CHANGE operations on it using functions.

Code & Input:

```
#include<stdio.h>
#define size 5
struct stack{
   int a[size],top;
   int temp[size], tos;
}s;
// Push operation....
void push(int item){
        s.a[++s.top] = item;
}
// Pop operation....
int pop(){
    return s.a[s.top--];
}
// Display operation....
void display(){
    int i;
    printf("\nThe stack contains: ");
    for(i = s.top; i>=0; i--){
        printf("\n\t%d", s.a[i]);
    }
}
// Peep operation....
void peep(){
    printf("\n\tTop : %d", s.top);
    printf("\n\tValue: %d",s.a[s.top]);
}
void change(int row, int new_element){
    int i;
    int j = -1;
    printf("\n\tTop: %d", s.top);
    for(i=s.top; i>row; i--){
        s.temp[++j] = s.a[s.top--];

    }
    s.a[s.top] = new_element;

    for(i = j; i>-1; i--){
        s.a[++s.top] = s.temp[j--];
```

```
    }
}
int main(){
    s.top = -1;
    int item, choice, row, new_element;
    char ans;
    do{
        printf("\n-----------------------------");
        printf("\nSTACK IMPLEMENTATION PROGRAM\n");
        printf("-----------------------------");
        printf("\n     1. Push\n     2. Pop\n     3. Display\n     4.
Peep\n     5. Change\n     6. Exit\n");
        printf("-----------------------------\n");
        printf("\n  Enter your choice: ");
        scanf("%d", &choice);
        switch(choice){
        case 1:
            if(s.top >= size-1){
                 printf("\nStack overflow..\n");
                 break;
            }
            printf("\nEnter item to be pushed:  ");
            scanf("%d", &item);
            push(item);
            break;
        case 2:
            if(s.top == -1){
                printf("\n..Stack underflow..\n");
                break;
            }
            pop();
            break;
        case 3:
            display();
            break;
        case 4:
            peep();
            break;
        case 5:
            printf("\n\tEnter row no : ");
            scanf("%d",&row);
            printf("\n\tEnter new element: ");
            scanf("%d", &new_element);
            change(row, new_element );
            break;
        case 6:
            return 0;
        }
    }while(choice != 6);
    return 0;
}
```

Screenshot of Code & Output:

main.c [LA2] - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

main.c

```c
#include<stdio.h>
#define size 5
struct stack{
    int a[size],top;
    int temp[size], tos;
}s;
// Push operation....
void push(int item)
{
        s.a[++s.top] = item;
}
// Pop operation....
int pop(){
    return s.a[s.top--];
}
// Display operation....
void display(){
    int i;
    printf("\nThe stack contains: ");
    for(i = s.top; i>=0; i--){
        printf("\n\t%d", s.a[i]);
    }
}
// Peep operation....
void peep(){
    printf("\n\tTop : %d", s.top);
    printf("\n\tValue: %d",s.a[s.top]);
}
void change(int row, int new_element){
    int i;
    int j = -1;
    printf("\n\tTop: %d", s.top);
    for(i=s.top; i>row; i--){
        s.temp[++j] = s.a[s.top--];
    /*
        display();
        printf("\n\tTop: %d", s.top);
        printf("\n\t j : %d", j);
    */
    }
    s.a[s.top] = new_element;
    /*
        display();
        printf("\n\tTop: %d", s.top);
        printf("\n\t j : %d", j);
    */
    for(i = j; i>-1; i--){
        s.a[++s.top] = s.temp[j--];
    /*
        display();
        printf("\n\tTop: %d", s.top);
        printf("\n\t j : %d", j);
    */
    }
}
int main(){

    s.top = -1;
    int item, choice, row, new_element,k;
    char ans;
    do{
        printf("\n-----------------------------");
        printf("\nSTACK IMPLEMENTATION PROGRAM\n");
        printf("-----------------------------");
        printf("\n     1. Push\n     2. Pop\n     3. Display\n     4. Peep\n     5.
        printf("-----------------------------\n");
        printf("\n  Enter your choice: ");
        scanf("%d", &choice);
```

```c
        switch(choice){
        case 1:
            if(s.top >= size-1){
                 printf("\nStack overflow..\n");
                 break;
            }
            printf("\nEnter the number of items  to be pushed:");
            scanf("%d",&k);
            printf("\nEnter items to be pushed:  ");
            for(int i=0;i<k;i++)
            {
            scanf("%d",&item);
            push(item);
            }
            break;
        case 2:
            if(s.top == -1)
            {
                printf("\n..Stack underflow..\n");
                break;
            }
            pop();
            break;
        case 3:
            display();
            break;
        case 4:
            peep();
            break;
        case 5:
            printf("\n\tEnter row no : ");
            scanf("%d",&row);
            printf("\n\tEnter new element: ");
            scanf("%d", &new_element);
            change(row, new_element );
            break;
        case 6:
            return 0;
        }
    }while(choice != 6);
    return 0;
}
```

Logs & others

Code::Blocks | Search results | Cccc | Build log | Build messages | CppCheck/Vera++ | CppChe

File	Line	Message
		=== Build: Debug in LA2 (compiler: GNU GCC Compiler) ===
C:\Users\Sh...		In function 'main':
C:\Users\Sh...	60	warning: unused variable 'ans' [-Wunused-variable]
		=== Build finished: 0 error(s), 1 warning(s) (0 minute(s), 2 second(s)) ===

C/C++ | Windows (CR+LF) | WINDOWS-1252 | Line 111, Col 1, Pos 2740 | Insert | Read/Write | default

ENG 05:16 PM

"C:\Users\Sharadindu\Desktop\DSA Lab Assignments\LA2\bin\Debug\LA2.exe"

```
----------------------------
STACK IMPLEMENTATION PROGRAM
----------------------------
     1. Push
     2. Pop
     3. Display
     4. Peep
     5. Change
     6. Exit
----------------------------

  Enter your choice: 1

Enter the number of items  to be pushed:5

Enter items to be pushed:  1
2
3
4
5
```

```
----------------------------
STACK IMPLEMENTATION PROGRAM
----------------------------
     1. Push
     2. Pop
     3. Display
     4. Peep
     5. Change
     6. Exit
----------------------------

  Enter your choice: 2

----------------------------
STACK IMPLEMENTATION PROGRAM
----------------------------
     1. Push
     2. Pop
     3. Display
     4. Peep
     5. Change
     6. Exit
----------------------------

  Enter your choice: 3

The stack contains:
        4
        3
        2
        1
----------------------------
STACK IMPLEMENTATION PROGRAM
----------------------------
     1. Push
     2. Pop
     3. Display
     4. Peep
     5. Change
     6. Exit
----------------------------

  Enter your choice: 4

        Top : 3
        Value: 4
----------------------------
STACK IMPLEMENTATION PROGRAM
----------------------------
     1. Push
     2. Pop
     3. Display
     4. Peep
     5. Change
     6. Exit
----------------------------

  Enter your choice: 5

        Enter row no : 3

        Enter new element: 7

        Top: 3
----------------------------
STACK IMPLEMENTATION PROGRAM
----------------------------
     1. Push
     2. Pop
     3. Display
     4. Peep
     5. Change
     6. Exit
----------------------------

  Enter your choice: 6

Process returned 0 (0x0)   execution time : 48.261 s
Press any key to continue.
```

Question 3

Problem:

WAP to convert the following expression to its postfix equivalent using stack:

a. ((A + B)*D) ^ (E – F)
b. A + (B*C – (D/E ^ F)*G)*H, where ^ denotes raise to the power.

Code:

```
#include<stdio.h>
#include<stdlib.h>
#include<ctype.h>
#include<string.h>
#define SIZE 100

char stack[SIZE];
int top = -1;
void push(char item)
{
  if(top >= SIZE-1)
  {
    printf("\nStack Overflow.");
  }
  else
  {
    top = top+1;
    stack[top] = item;
  }
}
char pop()
{
  char item ;
  if(top <0)
  {
    printf("stack under flow: invalid infix expression");
    getchar();
    /* underflow may occur for invalid expression */
    /* where ( and ) are not matched */
    exit(1);
  }
  else
  {
    item = stack[top];
    top = top-1;
    return(item);
  }
}
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 InfixToPostfix(char infix_exp[], char postfix_exp[])
{
  int i, j;
  char item;
  char x;
  push('(');
  strcat(infix_exp,")");
  i=0;
  j=0;
  item=infix_exp[i];
  while(item != '\0')
  {
    if(item == '(')
    {
      push(item);
    }
    else if( isdigit(item) || isalpha(item))
    {
      postfix_exp[j] = item;
      j++;
    }
    else if(is_operator(item) == 1)
    {
      x=pop();
      while(is_operator(x) == 1 && precedence(x)>= precedence(item))
      {
        postfix_exp[j] = x;
        j++;
        x = pop();
      }
```

```
      push(x);
      push(item);
    }
    else if(item == ')')
    {
      x = pop();
      while(x != '(')
      {
        postfix_exp[j] = x;
        j++;
        x = pop();
      }
    }
    else
    {
      printf("\nInvalid infix Expression.\n");
      getchar();
      exit(1);
    }
    i++;
    item = infix_exp[i];
  }
  if(top>0)
  {
    printf("\nInvalid infix Expression.\n");
    getchar();
    exit(1);
  }
  if(top>0)
  {
    printf("\nInvalid infix Expression.\n");
    getchar();
    exit(1);
  }
  postfix_exp[j] = '\0';

}
int main()
{
  char infix[SIZE], postfix[SIZE];
  printf("ASSUMPTION: The infix expression contains single letter
variables and single digit constants only.\n");
  printf("\nEnter Infix expression : ");
  gets(infix);
  InfixToPostfix(infix,postfix);
  printf("Postfix Expression: ");
  puts(postfix);
  return 0;
}
```

Inputs:

```
a. Enter Infix expression: ((A+B)*D)^(E-F)
b. Enter Infix expression: A+(B*C-(D/E^F)*G)*H
```

Outputs:

```
a. Postfix Expression: AB+D*EF-^
b. Postfix Expression: ABC*DEF^/G*-H*+
```

Screenshot of Code & CMD:

main.c [LA 2] - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

<global> | push(char item) : void

main.c

```c
#include<stdio.h>
#include<stdlib.h>
#include<ctype.h>
#include<string.h>
#define SIZE 100

char stack[SIZE];
int top = -1;
void push(char item)
{
  if(top >= SIZE-1)
  {
    printf("\nStack Overflow.");
  }
  else
  {
    top = top+1;
    stack[top] = item;
  }
}
char pop()
{
  char item ;
  if(top <0)
  {
    printf("stack under flow: invalid infix expression");
    getchar();
    /* underflow may occur for invalid expression */
    /* where ( and ) are not matched */
    exit(1);
  }
  else
  {
    item = stack[top];
    top = top-1;
    return(item);
  }
}
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 == '-')
  {
```

Management

Projects | Files | FSymbols | Resources

Workspace
LA 2
Sources
main.c

```
    return(1);
  }
  else
  {
    return(0);
  }
}
void InfixToPostfix(char infix_exp[], char postfix_exp[])
{
  int i, j;
  char item;
  char x;
  push('(');
  strcat(infix_exp,")");
  i=0;
  j=0;
  item=infix_exp[i];
  while(item != '\0')
  {
    if(item == '(')
    {
      push(item);
    }
    else if( isdigit(item) || isalpha(item))
    {
      postfix_exp[j] = item;
      j++;
    }
    else if(is_operator(item) == 1)
    {
      x=pop();
      while(is_operator(x) == 1 && precedence(x)>= precedence(item))
      {
        postfix_exp[j] = x;
        j++;
        x = pop();
      }
      push(x);
      push(item);
    }
    else if(item == ')')
    {
      x = pop();
      while(x != '(')
      {
        postfix_exp[j] = x;
        j++;
        x = pop();
      }
    }
    else
    {
      printf("\nInvalid infix Expression.\n");
      getchar();
      exit(1);
    }
    i++;
    item = infix_exp[i];
  }
  if(top>0)
  {
    printf("\nInvalid infix Expression.\n");
    getchar();
    exit(1);
  }
  if(top>0)
  {
    printf("\nInvalid infix Expression.\n");
    getchar();
    exit(1);
  }
  postfix_exp[j] = '\0';

}
int main()
{
  char infix[SIZE], postfix[SIZE];
  printf("ASSUMPTION: The infix expression contains single letter variables and single digit c
  printf("\nEnter Infix expression : ");
  gets(infix);
  InfixToPostfix(infix,postfix);
```

```c
    printf("Postfix Expression: ");
    puts(postfix);
    return 0;
}
```

Logs & others

Code::Blocks | Search results | Cccc | Build log | Build messages | CppCheck/Vera++ | CppCheck/Vera++ mess

```
Set variable: PATH=.;C:\Program Files\CodeBlocks\MinGW\bin;C:\Program Files\CodeBlocks\MinGW;C:\Program Files (x86)\Common Files
\Intel\Shared Libraries\redist\intel64\compiler;C:\Windows\System32;C:\Windows;C:\Windows\System32\wbem;C:\Windows\System32
\WindowsPowerShell\v1.0;C:\Windows\System32\OpenSSH;C:\Program Files (x86)\NVIDIA Corporation\PhysX\Common;C:\OrCAD\OrCAD_16.6
_Lite\tools\pspice;C:\OrCAD\OrCAD_16.6_Lite\tools\capture;C:\OrCAD\OrCAD_16.6_Lite\tools\bin;C:\OrCAD\OrCAD_16.6_Lite\OpenAccess
\bin\Win32\opt;C:\OrCAD\OrCAD_16.6_Lite\tools\fet\bin;C:\OrCAD\OrCAD_16.6_Lite\tools\pcb\bin;C:\Program Files\MiKTeX 2.9\miktex
\bin\x64;C:\Program Files\Git\cmd;C:\Program Files\nodejs;C:\Program Files\MATLAB\R2020a\runtime\win64;C:\Program Files\MATLAB
\R2020a\bin;C:\Users\Sharadindu\AppData\Local\Microsoft\WindowsApps;C:\Users\Sharadindu\AppData\Local\Programs\Microsoft VS Code
```

S C/C++ | Windows (CR+LF) | WINDOWS-1252 | Line 18, Col 23, Pos 275 | Insert | Read/Write | default

ENG 09:22 AM

```
"C:\Users\Sharadindu\Desktop\DSA Lab Assignments\Q3\bin\Debug\LA 2.exe"
ASSUMPTION: The infix expression contains single letter variables and single digit constants only.

Enter Infix expression : ((A+B)*D)^(E-F)
Postfix Expression: AB+D*EF-^

Process returned 0 (0x0)   execution time : 24.164 s
Press any key to continue.
```

```
"C:\Users\Sharadindu\Desktop\DSA Lab Assignments\Q3\bin\Debug\LA 2.exe"
ASSUMPTION: The infix expression contains single letter variables and single digit constants only.

Enter Infix expression : A+(B*C-(D/E^F)*G)*H
Postfix Expression: ABC*DEF^/G*-H*+

Process returned 0 (0x0)   execution time : 43.765 s
Press any key to continue.
```

Question 4

Problem:

Implement a stack S of n elements using arrays. Write functions to perform PUSH and POP operations. Implement queries using push and pop functions to

a. retrieve the mth element of the stack S from the top ($m < n$), leaving the stack without its top $m - 1$ elements.
b. retain only the elements in the odd position of the stack and pop out all even positioned elements.

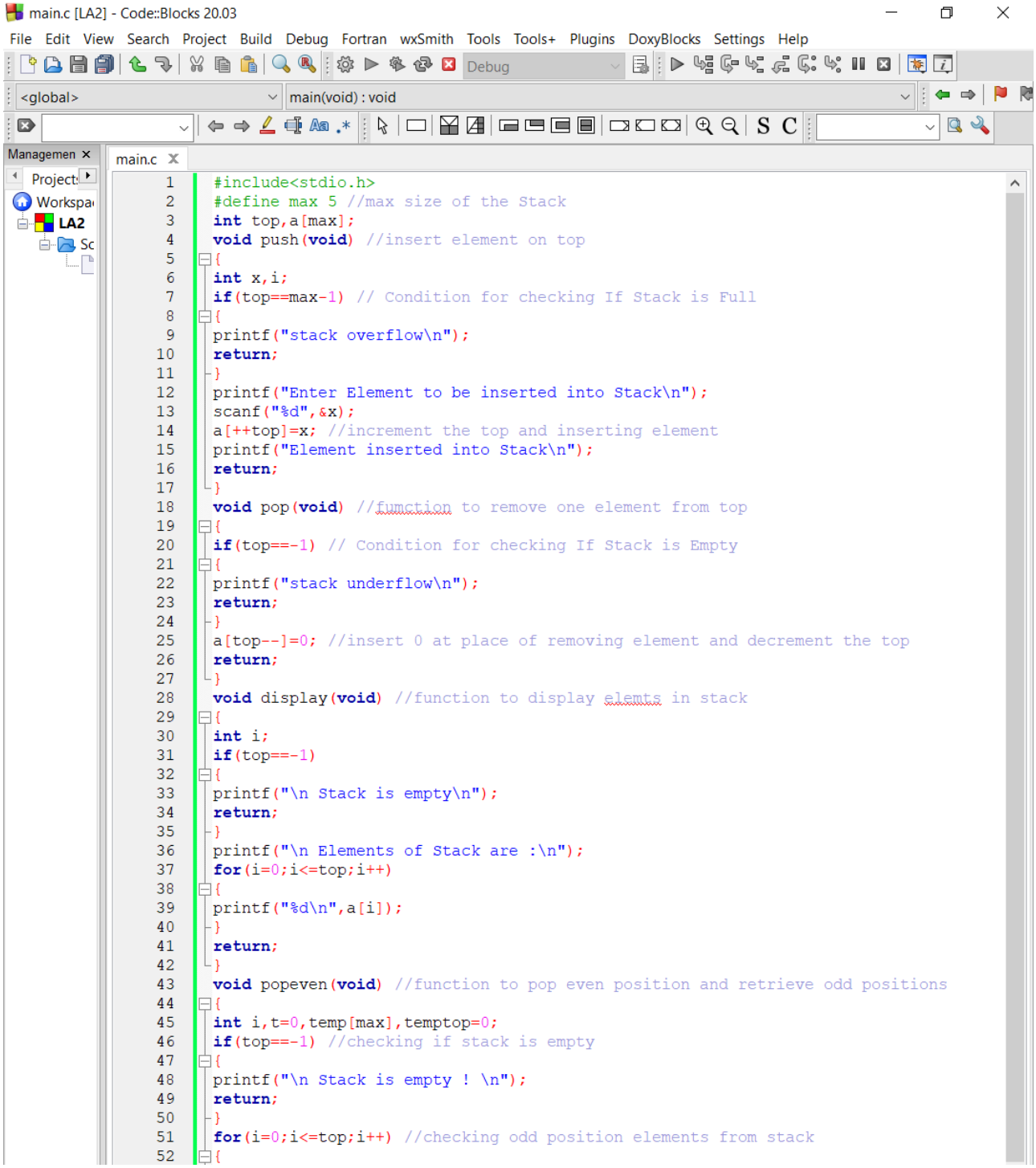
Code & Input:

```
#include<stdio.h>
#define max 5 //max size of the Stack
int top,a[max];
void push(void) //insert element on top
{
int x,i;
if(top==max-1) // Condition for checking If Stack is Full
{
printf("stack overflow\n");
return;
}
printf("Enter Element to be inserted into Stack\n");
scanf("%d",&x);
a[++top]=x; //increment the top and inserting element
printf("Element inserted into Stack\n");
return;
}
void pop(void) //fumction to remove one element from top
{
if(top==-1) // Condition for checking If Stack is Empty
{
printf("stack underflow\n");
return;
}
a[top--]=0; //insert 0 at place of removing element and decrement the
top
return;
}
void display(void) //function to display elemts in stack
{
int i;
if(top==-1)
{
printf("\n Stack is empty\n");
return;
}
printf("\n Elements of Stack are :\n");
for(i=0;i<=top;i++)
{
printf("%d\n",a[i]);
```

```
}
return;
}
void popeven(void) //function to pop even position and retrieve odd
positions
{
int i,t=0,temp[max],temptop=0;
if(top==-1) //checking if stack is empty
{
printf("\n Stack is empty ! \n");
return;
}
for(i=0;i<=top;i++) //checking odd position elements from stack
{
if((i+1)%2 != 0)
{
temp[t]=a[i];
t++;
}
}
temptop=top; //temporary variable to store value of top
for(i=0;i<=temptop;i++) //forloop to remove all elemnts from stack
{
pop();
}
for(i=0;i<=t;i++) //forloop to push only odd position elemts back into
stack
{
top++;
a[i]=temp[i];
}
printf("Elements of New Stack are :\n");
top--;
for(i=0;i<=top;i++) //display new stack
{
printf("%d\n",a[i]);
}
return;
}
void main(void) //main function
{
int c; top=-1;
system("cls");
do
{
printf("\n STACK OPERATIONS\n");
printf("\n  1:PUSH\n  2:POP  from  top(Mth  element)\n  3:POP  Even
positions\n 4:Display\n 5:Exit\n Choice:");
scanf("%d",&c);
switch(c)
{
case 1:push();printf("Maximum 5 elements !");
break;
case 2:pop();printf("Mth element poped! \n");
break;
case 3:popeven();
```

```
break;
case 4:display();
break;
case 5:printf("Program Ends\n");
break;
default :printf("Wrong Choice ! Enter Again !\n");
break;
}
}while(c!=5);
}
```

Screenshot of Code & Output:



```c
if((i+1)%2 != 0)
{
temp[t]=a[i];
t++;
}
}
temptop=top; //temporary variable to store value of top
for(i=0;i<=temptop;i++) //forloop to remove all elemnts from stack
{
pop();
}
for(i=0;i<=t;i++) //forloop to push only odd position elemts back into stack
{
top++;
a[i]=temp[i];
}
printf("Elements of New Stack are :\n");
top--;
for(i=0;i<=top;i++) //display new stack
{
printf("%d\n",a[i]);
}
return;
}
void main(void) //main function
{
int c; top=-1;
system("cls");
do
{
printf("\n STACK OPERATIONS\n");
printf("\n 1:PUSH\n 2:POP from top(Mth element)\n 3:POP Even positions\n 4:Display\
scanf("%d",&c);
switch(c)
{
case 1:push();printf("Maximum 5 elements !");
break;
case 2:pop();printf("Mth element poped! \n");
break;
case 3:popeven();
break;
case 4:display();
break;
case 5:printf("Program Ends\n");
break;
default :printf("Wrong Choice ! Enter Again !\n");
break;
}
}while(c!=5);
}
```

Logs & others

Code::Blocks | Search results | Cccc | Build log | Build messages | CppCheck/Vera++ | CppChe

```
Set variable: PATH=.;C:\Program Files\CodeBlocks\MinGW\bin;C:\Program Files\CodeBlocks\MinGW;C:\Program Files
(x86)\Common Files\Intel\Shared Libraries\redist\intel64\compiler;C:\Windows\System32;C:\Windows;C:\Windows\System32
\wbem;C:\Windows\System32\WindowsPowerShell\v1.0;C:\Windows\System32\OpenSSH;C:\Program Files (x86)\NVIDIA
Corporation\PhysX\Common;C:\OrCAD\OrCAD_16.6_Lite\tools\pspice;C:\OrCAD\OrCAD_16.6_Lite\tools\capture;C:\OrCAD
\OrCAD_16.6_Lite\tools\bin;C:\OrCAD\OrCAD_16.6_Lite\OpenAccess\bin\Win32\opt;C:\OrCAD\OrCAD_16.6_Lite\tools\fet
\bin;C:\OrCAD\OrCAD_16.6_Lite\tools\pcb\bin;C:\Program Files\MiKTeX 2.9\miktex\bin\x64;C:\Program Files\Git\cmd;C:
\Program Files\nodejs;C:\Program Files\MATLAB\R2020a\runtime\win64;C:\Program Files\MATLAB\R2020a\bin;C:\Users
```

"C:\Users\Sharadindu\Desktop\DSA Lab Assignments\LA2\bin\Debug\LA2.exe"

```
 STACK OPERATIONS

 1:PUSH
 2:POP from top(Mth element)
 3:POP Even positions
 4:Display
 5:Exit
 Choice:1
Enter Element to be inserted into Stack
3
Element inserted into Stack
Maximum 5 elements !
 STACK OPERATIONS

 1:PUSH
 2:POP from top(Mth element)
```

```
 3:POP Even positions
 4:Display
 5:Exit
 Choice:1
Enter Element to be inserted into Stack
7
Element inserted into Stack
Maximum 5 elements !
 STACK OPERATIONS

 1:PUSH
 2:POP from top(Mth element)
 3:POP Even positions
 4:Display
 5:Exit
 Choice:1
Enter Element to be inserted into Stack
9
Element inserted into Stack
Maximum 5 elements !
 STACK OPERATIONS

 1:PUSH
 2:POP from top(Mth element)
 3:POP Even positions
 4:Display
 5:Exit
 Choice:1
Enter Element to be inserted into Stack
10
Element inserted into Stack
Maximum 5 elements !
 STACK OPERATIONS

 1:PUSH
 2:POP from top(Mth element)
 3:POP Even positions
 4:Display
 5:Exit
 Choice:4

 Elements of Stack are :
3
7
9
10

 STACK OPERATIONS

 1:PUSH
 2:POP from top(Mth element)
 3:POP Even positions
 4:Display
 5:Exit
 Choice:3
Elements of New Stack are :
3
9

 STACK OPERATIONS

 1:PUSH
 2:POP from top(Mth element)
 3:POP Even positions
 4:Display
 5:Exit
 Choice:2
Mth element poped!

 STACK OPERATIONS

 1:PUSH
 2:POP from top(Mth element)
 3:POP Even positions
 4:Display
```

```
 5:Exit
 Choice:4

 Elements of Stack are :
3
```

Question 5

Problem:

Implement a program to evaluate any given postfix expression. Test your program for the evaluation of the equivalent postfix form of the expression:

(– (A*B) / D) ↑ C + E – F * H * I for A = 1, B = 2, D = 3, C = 14, E = 110, F = 220, H = 16.78, I = 364.621.

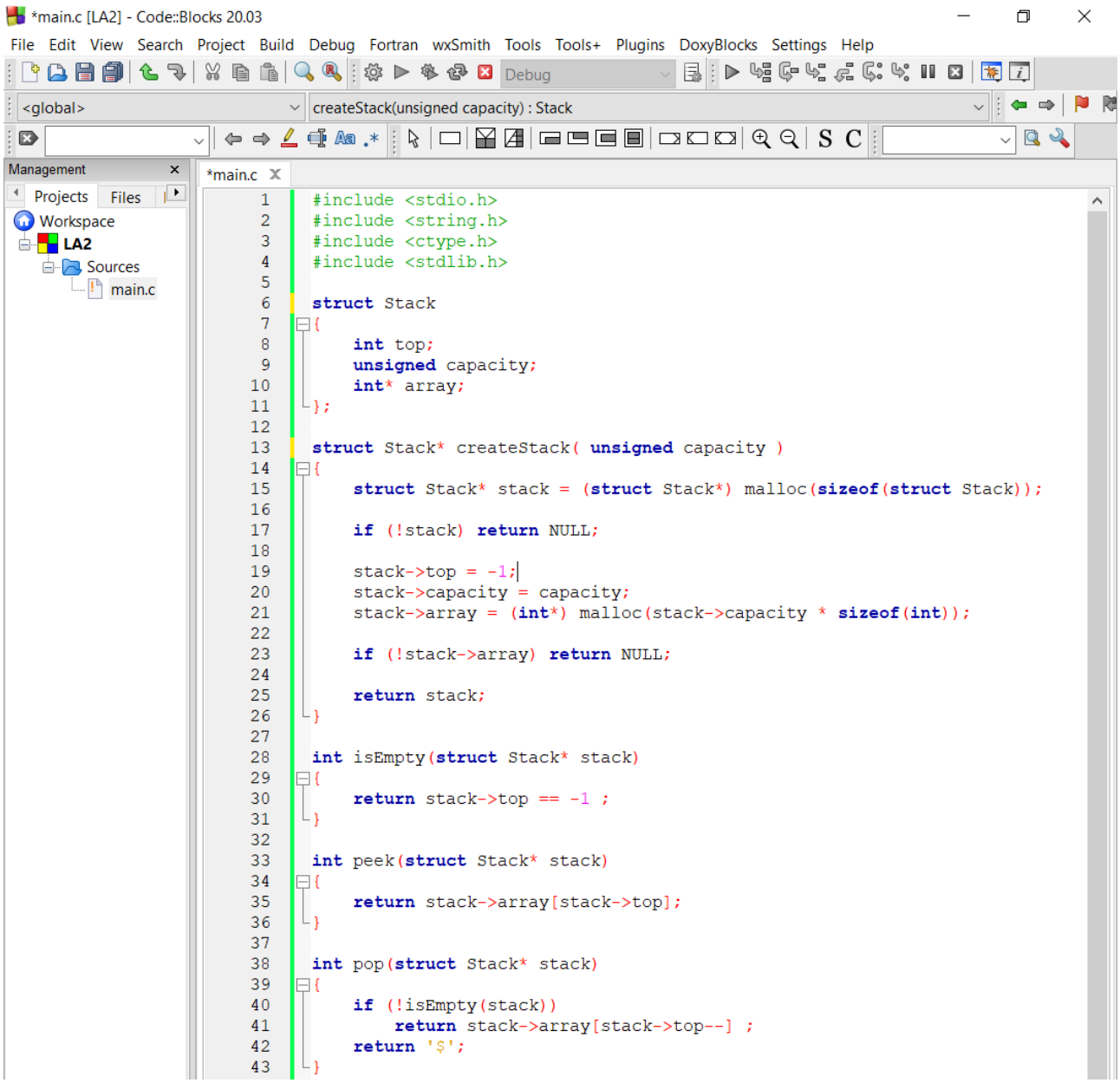
Code & Input:

```
#include <stdio.h>
#include <string.h>
#include <ctype.h>
#include <stdlib.h>

// Stack type
struct Stack
{
    int top;
    unsigned capacity;
    int* array;
};

// Stack Operations
struct Stack* createStack( unsigned capacity )
{
    struct Stack* stack = (struct Stack*) malloc(sizeof(struct Stack));

    if (!stack) return NULL;

    stack->top = -1;
    stack->capacity = capacity;
    stack->array = (int*) malloc(stack->capacity * sizeof(int));

    if (!stack->array) return NULL;

    return stack;
}

int isEmpty(struct Stack* stack)
{
    return stack->top == -1 ;
}

int peek(struct Stack* stack)
{
    return stack->array[stack->top];
}

int pop(struct Stack* stack)
{
    if (!isEmpty(stack))
        return stack->array[stack->top--] ;
```

```
    return '$';
}

void push(struct Stack* stack,int op)
{
    stack->array[++stack->top] = op;
}


// The main function that returns value
// of a given postfix expression
int evaluatePostfix(char* exp)
{
    // Create a stack of capacity equal to expression size
    struct Stack* stack = createStack(strlen(exp));
    int i;

    // See if stack was created successfully
    if (!stack) return -1;

    // Scan all characters one by one
    for (i = 0; exp[i]; ++i)
    {
        //if the character is blank space then continue
        if(exp[i]==' ')continue;

        // If the scanned character is an
        // operand (number here),extract the full number
        // Push it to the stack.
        else if (isdigit(exp[i]))
        {
            int num=0;

            //extract full number
            while(isdigit(exp[i]))
            {
            num=num*10 + (int)(exp[i]-'0');
                i++;
            }
            i--;

            //push the element in the stack
            push(stack,num);
        }

        // If the scanned character is an operator, pop two
        // elements from stack apply the operator
        else
        {
            int val1 = pop(stack);
            int val2 = pop(stack);

            switch (exp[i])
            {
            case '+': push(stack, val2 + val1); break;
            case '-': push(stack, val2 - val1); break;
```

```
            case '*': push(stack, val2 * val1); break;
            case '/': push(stack, val2/val1); break;

            }
        }
    }
    return pop(stack);
}

// Driver program to test above functions
int main()
{
    char exp[] = "(-(1*2)/3)^14+110-220*16.78*346.621";
    printf ("%d", evaluatePostfix(exp));
    return 0;
}
```

Screenshot of Code & Output:



```c

void push(struct Stack* stack,int op)
{
    stack->array[++stack->top] = op;
}

int evaluatePostfix(char* exp)
{
    struct Stack* stack = createStack(strlen(exp));
    int i;

    if (!stack) return -1;

    for (i = 0; exp[i]; ++i)
    {
        if(exp[i]==' ')continue;

        else if (isdigit(exp[i]))
        {
            int num=0;

            while(isdigit(exp[i]))
            {
            num=num*10 + (int)(exp[i]-'0');
                i++;
            }
            i--;

            push(stack,num);
        }

        else
        {
            int val1 = pop(stack);
            int val2 = pop(stack);

            switch (exp[i])
            {
            case '+': push(stack, val2 + val1); break;
            case '-': push(stack, val2 - val1); break;
            case '*': push(stack, val2 * val1); break;
            case '/': push(stack, val2/val1); break;

            }
        }
    }
    return pop(stack);
}

int main()
{
    char exp[] = "(-(1*2)/3)^14+110-220*16.78*346.621";
    printf ("%d", evaluatePostfix(exp));
    return 0;
}

```

Logs & others

Code::Blocks | Search results | Cccc | Build log | Build messages | CppCheck/Vera++

```
Set variable: PATH=.;C:\Program Files\CodeBlocks\MinGW\bin;C:\Program Files\CodeBlocks\MinGW;C:\Program
Files (x86)\Common Files\Intel\Shared Libraries\redist\intel64\compiler;C:\Windows\System32;C:\Windows;C:
\Windows\System32\wbem;C:\Windows\System32\WindowsPowerShell\v1.0;C:\Windows\System32\OpenSSH;C:\Program
Files (x86)\NVIDIA Corporation\PhysX\Common;C:\OrCAD\OrCAD_16.6_Lite\tools\pspice;C:\OrCAD\OrCAD_16.6_Lite
\tools\capture;C:\OrCAD\OrCAD_16.6_Lite\tools\bin;C:\OrCAD\OrCAD_16.6_Lite\OpenAccess\bin\Win32\opt;C:
\OrCAD\OrCAD_16.6_Lite\tools\fet\bin;C:\OrCAD\OrCAD_16.6_Lite\tools\pcb\bin;C:\Program Files\MiKTeX 2.9
\miktex\bin\x64;C:\Program Files\Git\cmd;C:\Program Files\nodejs;C:\Program Files\MATLAB\R2020a\runtime
```

C/C++ | Windows (CR+LF) | WINDOWS-1252 | Line 19, Col 21, Pos 340 | Insert | Modified | Read/Write | default

ENG 07:05 PM

"C:\Users\Sharadindu\Desktop\DSA Lab Assignments\LA2\bin\Debug\LA2.exe"

```
621
Process returned 0 (0x0)   execution time : 0.083 s
Press any key to continue.
```