Question 01

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

int val;
int top,max;
int stack[100];

int main()
{
   int getSize();
   int isEmpty(int stack[]);
   int push(int stack[],int val);
   int pop(int stack[]);
   int peek(int stack[]);
   int isfull(int stack[]);
   int printstack(int stack[]);

 return 0;
}



int isEmpty(int stack[])
{
   if(top==-1)
   {
      return 1;
```

```
    }
    else
    {
        return 0;
    }
}

int push(int stack[],int val)
{
    if(top=max-1)
    {
        printf("\nstack is overflow.push in impossible");
    }
    else
    {
      stack[top]=val;
      top++;
      return stack[top];
    }

}

 int pop(int stack[])
   {
       if(top=-1)
        {
            printf("\nstack is underflow.pop is impossible");
```

```
      }
      else{
         stack[top]=val;
         top--;
         return stack[top];
      }
  }

 int isfull(int stack[])
{
    if(top=max-1)
    {
       return 1;
    }
    else
    {
       return 0;
    }
}

int display(int stack[])
  {
      int i;
      if(top==-1)
      {
         printf("\nNo values.stack is empty.");
      }
```

```
      else
      {
      printf("\n Elements of the stack : \n");
      for(i=top;i>=0;i--)
      {
         printf("\n%d",stack[i]);
      }
      }
   }

int peek(int stack[])
{
    if (top==-1)
    {
        printf("\nstack is underflow");
    }
    else
    {
    stack[top]=val;
    return stack[top];
    }
}


int getSize()
    {
      int i;
```

```
    printf ("\n\nThe size of stack is %d\n",top+1);
  }
```

Question 02

```
#include<stdio.h>
#include<stdlib.h>
#include<conio.h>

#define  max 50

int val;
int top,stack[max];

int main()
{
  int peek();
  printf("PEEK : %d",peek());
  return 0;
}

int peek()
{
  if (top==-1)
  {
     printf("\nstack is underflow");
  }
```

```
    else
    {
    stack[top]=val;
    return stack[top];
    }
}
```

Question 03

```
#include<stdio.h>
#include<conio.h>
#define max 100

int stack[100];
int val;
int top;



int isEmpty();
int push(int);
int pop();
int peek();
int isfull();
int display();
int getSize();
```

```
int main()
{
   isEmpty();
   push(34);
   push(56);
   pop();
   isEmpty();
   peek();
   push(23);
   isfull();
   peek();
   push(73);
   display();
 return 0;
}


int isEmpty()
{
   if(top==-1)
   {
      return 1;
   }
   else
   {
      return 0;
   }
```

```
}

int push(int val)
{
   if(top==max-1)
   {
      printf("\nstack is overflow.push in impossible");
   }
   else
   {
    stack[top]=val;
    top++;
    return stack[top];
   }

}

 int pop()
  {
     if(top==-1)
      {
         printf("\nstack is underflow.pop is impossible");
      }
      else{
         stack[top]=val;
         top--;
         return stack[top];
```

```
      }
  }

int isfull()
{
   if(top==max-1)
   {
      return 1;
   }
   else
   {
      return 0;
   }
}

int display()
  {
     int i;
     if(top==-1)
     {
        printf("\nNo values.stack is empty.");
     }
     else
     {
     printf("\n Elements of the stack : \n");
     }
     for(i=top;i>=0;i--)
```

```
      {
         printf("\n%d",stack[i]);

      }
   }

int peek()
{
   if (top==-1)
   {
      printf("\nstack is underflow");
   }
   else
   {
   stack[top]=val;
   return stack[top];
   }
}

   int getSize()
   {
     int i;
     printf ("\n\nThe size of stack is %d\n",top+1);
   }
```

Question 04

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

#define max 50
int top,stack[max];

void push(char val){

    if(top == max-1)
    {
       printf("stack overflow");
    }
    else
    {
       stack[++top]=val ;
    }

}

void pop()
{
    printf("%c",stack[top--]);
}


int main()
{
```

```
  char str[50];
  printf("Enter your string : ");
  gets(str);
  int len = strlen(str);
  int i;

  for(i=0;i<len;i++)
  {
      push(str[i]);
  }

 printf("\nReverse string....\n");
  for(i=0;i<len;i++)
  {
    pop();
  }
  return 0;
}
```

Question 05

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

#define MAX 100
```

```
int top=-1;
int stack[MAX];

int isFull() {
   if(top >= MAX-1)
      return 1;
   else
      return 0;
}

int isEmpty() {
 if(top == -1)
    return 1;
 else
    return 0;
}

int push() {
   if (isFull())
      printf("Stack is overflow.push is impossible.\n");
   else {
       return stack[top++];
   }
}

int pop() {
```

```
    if (isEmpty())
        printf("Stack is underflow.pop is impossible.\n");
    else {
     top = top - 1;
        return stack[top];
    }
}

int main()
 {
    char str[100];
    int i, j,len;
    printf("Enter a string : ");
    gets(str);

    len = strlen(str);

    for(i=0; i<len; i++)
    {
        push(str[i]);
    }

    for(j=top;j>=0;j--)
    {
        if(pop()!=str[i])
        {
            printf("\nThis string is not a Palindrome String\n\n");
```

```
        return 0;
      }
   }

   printf("\nThis string is a Palindrome String\n\n");
   return 0;

}
```

Question 06

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

#define max 100

char stack[max];
int top = -1;

void push(char val)
{
   stack[++top] = val;
}
```

```
char pop()
{
   if(top == -1)
      return -1;
   else
      return stack[top--];
}

int priority(char val)
{
   if(val == '(')
      return 0;
   if(val == '+' || val == '-')
      return 1;
   if(val== '*' || val == '/')
      return 2;
   return 0;
}

int main()
{
   char stack[max];
   char *x, val;
   printf("Enter the expression : ");
   scanf("%s",stack);
   printf("\n");
   x = stack;
```

```
    while(*x != '\0')
    {
        if(isalnum(*x))
            printf("%c ",*x);
        else if(*x == '(')
            push(*x);
        else if(*x == ')')
        {
            while((val = pop()) != '(')
                printf("%c ", val);
        }
        else
        {
            while(priority(stack[top]) >= priority(*x))
                printf("%c ",pop());
            push(*x);
        }
        x++;
    }

    while(top != -1)
    {
        printf("%c ",pop());
    }
return 0;
}
```

Question 06

(A+B)*(C+D)

Infix character scanned	Stack	Postfix Expression
(	(	
A	(	A
+	(+	A
B	(+	AB
)	(+)	AB+
*	*	AB+
(	(	AB+*
C	(	AB+*C
+	(+	AB+*C
D	(+	AB+*CD
)	(+)	AB+*CD
		AB+*CD+

(A+B)*C

Infix character scanned	Stack	Postfix Expression
(	(	
A	(	A
+	(+	A
B	(+	AB
)	(+)	AB+
*	*	AB+
C	*	AB+C
		AB+C*

A+B*C

Infix character scanned	Stack	Postfix Expression
A		A
+	+	A

B	+	AB
*	+*	AB
C	+*	ABC
		ABC*+

3+4*5/6

Infix character scanned	Stack	Postfix Expression
3		3
+	+	3
4	+	3 4
*	+*	3 4
5	+*	3 4 5
/	+*	3 4 5 /
6	+*	3 4 5 /
		3 4 5 /*+

Question 07

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

#define MAX 100

int top=-1;
int stack[MAX];

int isFull() {
   if(top >= MAX-1)
```

```
      return 1;
   else
      return 0;
}

int isEmpty() {
 if(top == -1)
    return 1;
 else
    return 0;
}

int push() {
   if (isFull())
      printf("Stack is overflow.push is impossible.\n");
   else {
       return stack[top++];
   }
}

int pop() {
   if (isEmpty())
      printf("Stack is underflow.pop is impossible.\n");
   else {
    top = top - 1;
      return stack[top];
   }
}
```

```
int peek()
{
   if (top==-1)
   {
      printf("\nstack is underflow");
   }
   else
   {
      return stack[top];
   }
}

int main()
 {
   char str[100];
   int i, len,x;
   printf("Enter a string with paranthesis : ");
   scanf("%c", &str);

   len = strlen(str);

   for(i=0; i<len; i++)
 {
    if(str[i]== '{' || str[i]=='[' || str[i]=='(' )
      {
         push(str[i]);
         continue;
      }
   else if(str[i] == '}' || str[i]==']' || str[i]==')' )
```

```
{
   if(str[i]=='}')
   {
      if(peek()=='{')
      {
         pop();
      }
      else
      {
         x=1;
         break;
      }
   }

  else if(str[i]==']')
   {
      if(peek()=='[')
      {
         pop();
      }
      else
      {
         x=1;
         break;
      }
   }

  else if(str[i]==')')
   {
```

```
            if(peek()=='(')
            {
               pop();
            }
            else
            {
               x=1;
               break;
            }
        }
     }

   }


   if(x!=1)
   {
      printf("\n Valid  Expression\n");
   }
   else
   {
     printf("\n InValid  Expression\n");
   }

return 0;
 }
```

Question 08

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

int toh(int n,char beg,char aux,char end);

int main()
{
  int num;
  printf("Enter the number of discs : ");
  scanf("%d", &num);
  printf("\n");
  toh(num,'A','B','C');

  return 0;
}

int toh(int n,char beg,char aux,char end)
{
    if(n>=1)
    {
    toh(n-1,beg,end,aux);
    printf("%d disk move %c to %c.\n",n,beg,end);
    toh(n-1,aux,beg,end);
    }
}
```

Question 09

```
#include<stdio.h>

int gcd(int a,int b);

int main()
{
   int num1,num2;

   printf("Enter two numbers...\n");
   printf("Number 1 : ");
   scanf("%d", & num1);
   printf("Number 2 : ");
   scanf("%d", & num2);

   int res=gcd(num1,num2);
   printf("%d", res);

return 0;

}

int gcd(int a,int b)
{
   if(a>b)
 {
   if(b==0)
```

```
    {
        return (a);
    }
    return gcd(b,a%b);
}

if(a<b)
    {
    if(a==0)
    {
        return (b);
    }
    return gcd(a,b%a);
    }
}
```