Name :- Kunj Trivedi
Roll No :- 38
SE - 9
Batch :- C

Assignment 2A

(1)
```
#include <stdio.h>
# include<string.h>
# include MAXSIZE 1000

Struct stack {
        int top ;
        int arry [MAXSIZE];
    } st;

    int isFull () {
      if (st.top >= MAXSIZE - 1)
            return 1 ;
      else
            return 0;
    }
    int   isEmpty () {
     if   (st.top == -1)
          return 1;
    else
          return 0;
    void  push (int num) {
        int (isfull ())
              printf (" Stack is full ...\n");
        else {
              st.array[st.top + 1] = num;
              st. top ++ ;
          }
    }
```

Kuny
38

```
int pop(){
    if (isEmpty ())
        printf ("stack is Empty ...\n");
    else {
        st.top = st.top - 1.
        return st.array [st.top + 1];
    }
}
// Initializing top index to -1
void initialize () {
    st.top = -1;
}

int main() {
    char input string [100], c;
    int i, length;
    initialize ();
    printf (" Enter a string of parenthesis ...\n");
    gets (input string).
    length = strlen (input string).
    for (i = 0. i < length; i++) {
        if (input string [i] == '{')
            push (inputstring [i]);
        else if ( input string [i] == '}').
            pop ();
        else {
            printf ("Error : Invalid Character!! \n");
            return 0;
        }
    }
    if (isEmpty())
        printf ("Valid parantheses Expression \n");
```

Kunj
38

Camlin Page
Date / /

```
    else
        printf ("Invalid parantheis Expression \n");
    return 0;
}
```

Output :

```
Enter a string of paranthesies
{{ {}{ }}}
Valid paranthesis Expression
```

Q. 2]

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

Void EnqueueRear (int);
int dequeueFront ();
int dequeueRear ();
Void display ().
int queue [SIZE];
int  rear = 0, front = 0;
int main ()
{
    Char ch;
    int choice 2, value;
    {
```

Kunj
38

```
    printf ("\n Select the operation \n");
    printf ("1.Insert \n2. Delete from Rear \n3. Delete
              from front \n4. display \n5. Exit ");
  do
   {
     printf ("\n Enter your choice :");
     scanf ("%d", & choice2);
    switch (choice2)
     {
        case 1 : enqueueRear (value);
                 display ();
                 break
        case 2 : value = dequeueRear ().
                 printf (" \n The deleted value is
                         %d ", value);
                 display () ;
                 break;
       case 3 : value = dequeueFront ().
                printf ("\n The value deleted is
                          %d ", value);
                display ();
                break;
       case 4 : display ();
                break;
       case 5 : exit (0);
       default : printf (" Wrong choice ");
     }
  } while (choice2! =5);
   getch ();
  }
}
```

Kunj
38

Camlin Page
Date / /

```
void enqueueRear(int value)
{
  char ch;
  if (front== SIZE (2)
   {
      printf ("\n Queue is full !!!");
      return;
   }
   do
   {
      printf ("\n Enter the value to be insert
      scanf ("%d"& value);
       queue [front] = value;
        front ++;
        printf ("Do you want to continue
                insertion Y/N");
            ch = getch ();
      } while (ch=='y' or ch=='Y');
 }
 int dequeueRear()
 {
      int delete;
      if (front==rear)
      {
         printf ("\n Queue is Empty");
         return 0;
      }
        rear ++;
        delete = queue [rear-1];
         return delete;
    }
```

Kunj
38

```
void display ()
{
    int i;
    if (front= =rear)
        printf ("\n Queue is Empty");
    else {
        printf ("\n The Queue elements are:");
        for (i =rear; i < front; i++)
        {
            printf (" %d \t", queue[i]);
        }
    }
}
```