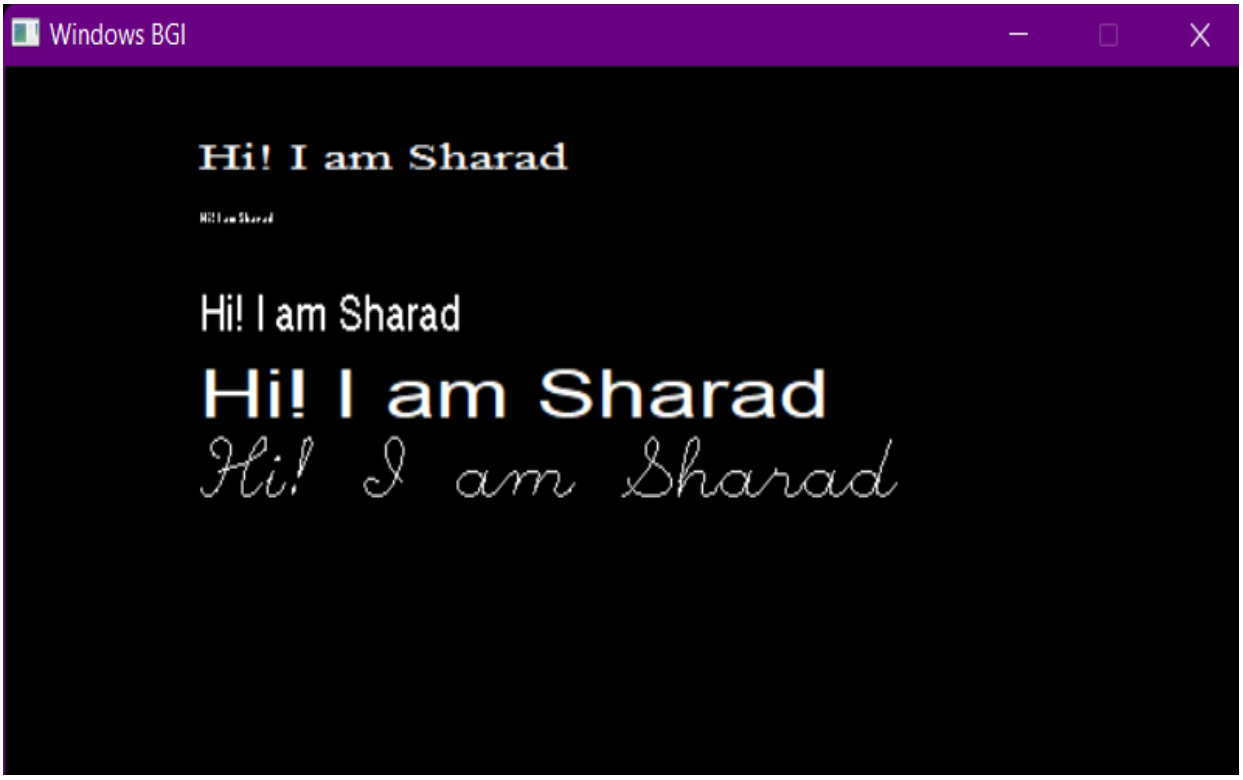
Lab No: 1

WRITE A PROGRAM TO MANIPULATE TEXT ON C/C++ USING STANDARD GRAPHICS FUNCTIONS

Source Code:

```
#include<graphics.h>
#include<conio.h>
int main(){
int gd= DETECT, gm;
initgraph(&gd,&gm,"");
for (int i=1;i<=5;i++)
{
//setting color for text
setcolor (WHITE);//setcolor (int color) ;
//changing fontface, orientation and size;
settextstyle(i,0,i);//settextstyle(int font, int orientation, int size);
//print the desired message.
outtextxy (100, 30*i, "HELLO WORLD!"); //outtextxy(int x, int y, char value)
}
getch();
closegraph();
return 0;
}
```

Output:


Windows BGI
Hi! I am Sharad
Hi! I am Sharad
Hi! I am Sharad
Hi! I am Sharad

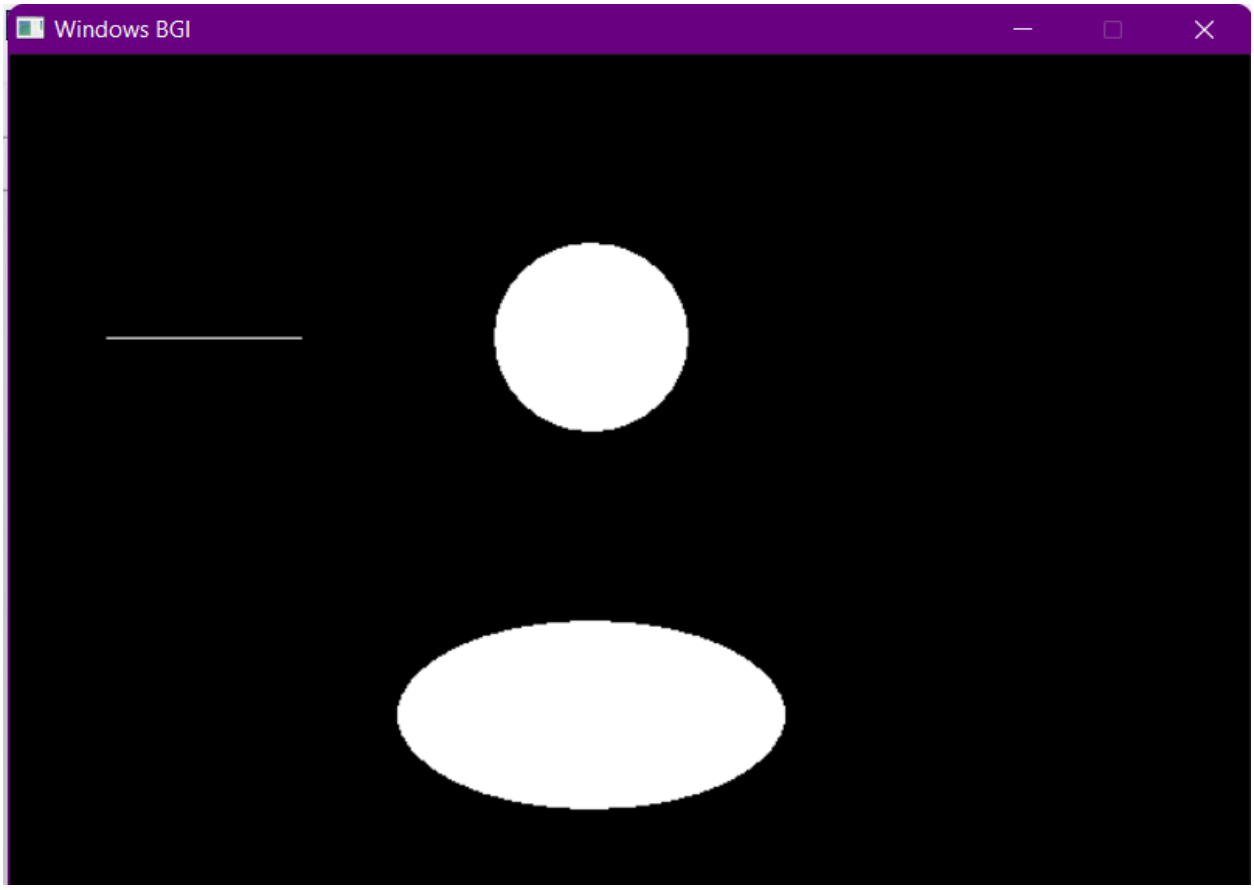
Lab No: 2

WRITE A PROGRAM TO DRAW GRAPHICS PRIMITIVES USING STANDARD FUNCTION.

Source Code:

```
#include<graphics.h>
#include<conio.h>
int main(){
   int gd= DETECT, gm;
   initgraph(&gd, &gm,"");
   //Setting filling style
   setfillstyle(SOLID_FILL,WHITE) ;//setfillstyle(int pattern, int color);
   //Drawing circle
   circle(300,150,50) ; //circle(int x, int y, int radius);
   //Filling color inside a boundary.
   floodfill(301,150, WHITE); //floodfill(int x , int y, int bordercolor);
   setfillstyle(SOLID_FILL,WHITE) ;
   //drawing ellipse
   ellipse(300,350,0,360,100,50) ; //elLipse(int x, int y, int startangle, int endangle, int
xradius, int yradius);
   floodfill(301,350,WHITE);
   //setting color
   setcolor (WHITE) ;//setcolor (int color)
   //line drawing
   line (150,150,50,150) ;
   //Line (int x1, int y1, int x2, int y2)
   getch();
   closegraph();
   return 0;
}
```

Output:



Lab No: 3

WRITE A PROGRAM TO ANIMATE A CAR USING GRAPHICS FUNCTIONS.

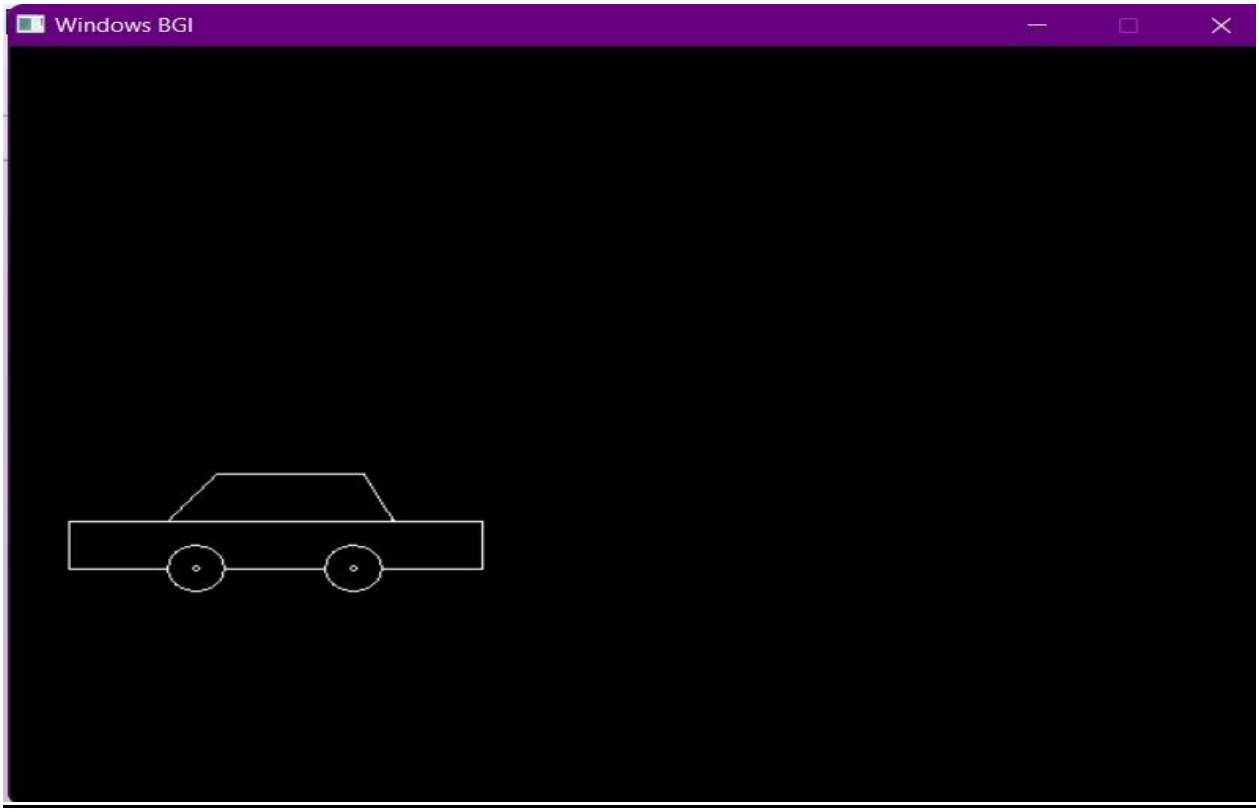
Source Code:

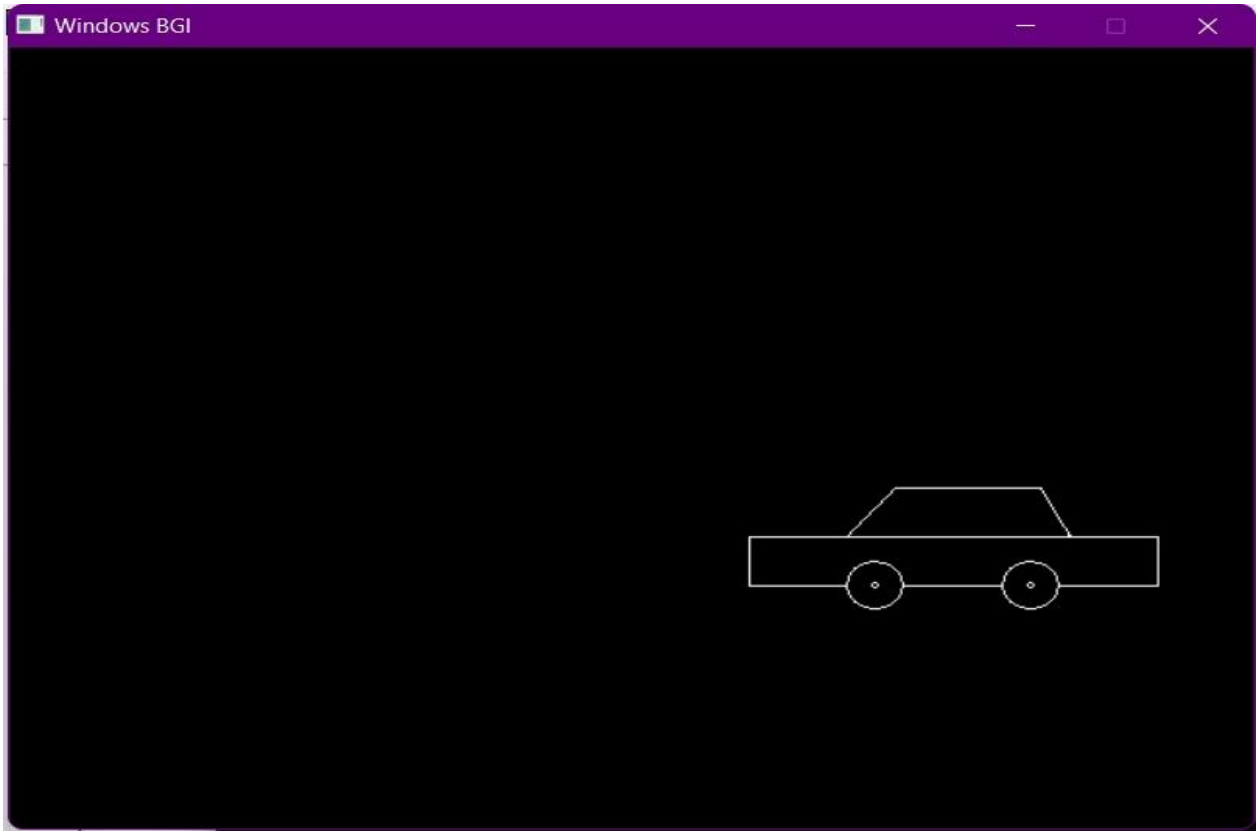
```
#include <graphics.h>
#include <conio.h>
void draw_moving_car(void) {
   int i, j = 0, gd = DETECT, gm;
   initgraph(&gd, &gm, "");
   for (i = 0; i <= 420; i = i + 10) {
      // Set color of car as white
      setcolor(WHITE);
      // body of car
      line(0 + i, 300, 210 + i, 300);
      line(50 + i, 300, 75 + i, 270);
      line(75 + i, 270, 150 + i, 270);
      line(150 + i, 270, 165 + i, 300);
      line(0 + i, 300, 0 + i, 330);
      line(210 + i, 300, 210 + i, 330);
      // For left wheel of car
      circle(65 + i, 330, 15);
      circle(65 + i, 330, 2);
      // For right wheel of car
      circle(145 + i, 330, 15);
      circle(145 + i, 330, 2);
      // Line left of left wheel
      line(0 + i, 330, 50 + i, 330);
      // Line middle of both wheel
      line(80 + i, 330, 130 + i, 330);
      // Line right of right wheel
      line(210 + i, 330, 160 + i, 330);
      delay(100);

      // To erase previous drawn car, draw the whole car at same position but using black
color
      setcolor(BLACK);
      // Lines for bonnet and body of car
      line(0 + i, 300, 210 + i, 300);
      line(50 + i, 300, 75 + i, 270);
```

```
        line(75 + i, 270, 150 + i, 270);
        line(150 + i, 270, 165 + i, 300);
        line(0 + i, 300, 0 + i, 330);
        line(210 + i, 300, 210 + i, 330);
        // For left wheel of car
        circle(65 + i, 330, 15);
        circle(65 + i, 330, 2);
        // For right wheel of car
        circle(145 + i, 330, 15);
        circle(145 + i, 330, 2);
        // Line left of left wheel
        line(0 + i, 330, 50 + i, 330);
        // Line middle of both wheel
        line(80 + i, 330, 130 + i, 330);
        // Line right of right wheel
        line(210 + i, 330, 160 + i, 330);
    }
    getch();
    closegraph();
}
int main()
{
    draw_moving_car();
    return 0;
}
```

Output:


Windows BGI


Windows BGI

Lab No: 4

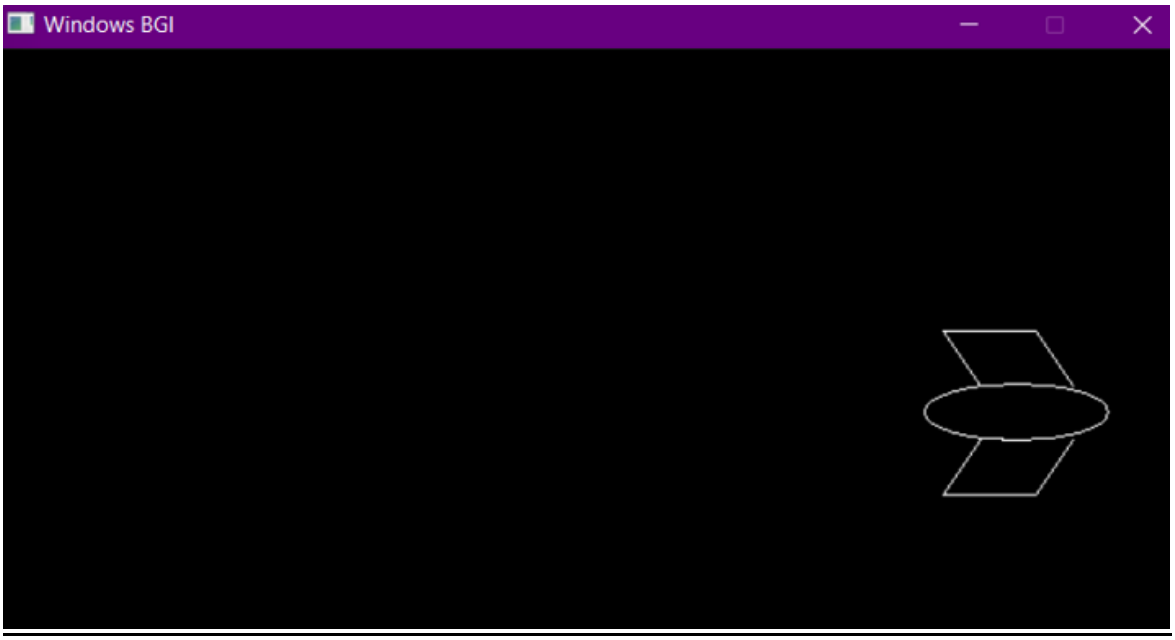
WRITE A PROGRAM TO ANIMATE A SIMPLE AEROPLANE USING GRAPHICS FUNCTION.

Source Code:

```
#include<graphics.h>
int main(){
        int gd = DETECT, gm, i;
        initgraph(&gd,&gm,"");
        for (i=10;1<=620; i++)
        {
                cleardevice();
                setcolor(WHITE);
                ellipse (100+i,200,0,360,50,16);
                line (130+i, 185,110+i,155),
                line (110+i,155,60+i,155),
                line (60+i,155,80+i,185) ,
                line (80+i, 215,60+i,245),
                line (60+i,245,110+i,245) ;
                line(110+i,245,130+i,215);
                delay (10);
        }
}
```

Output:


Windows BGI

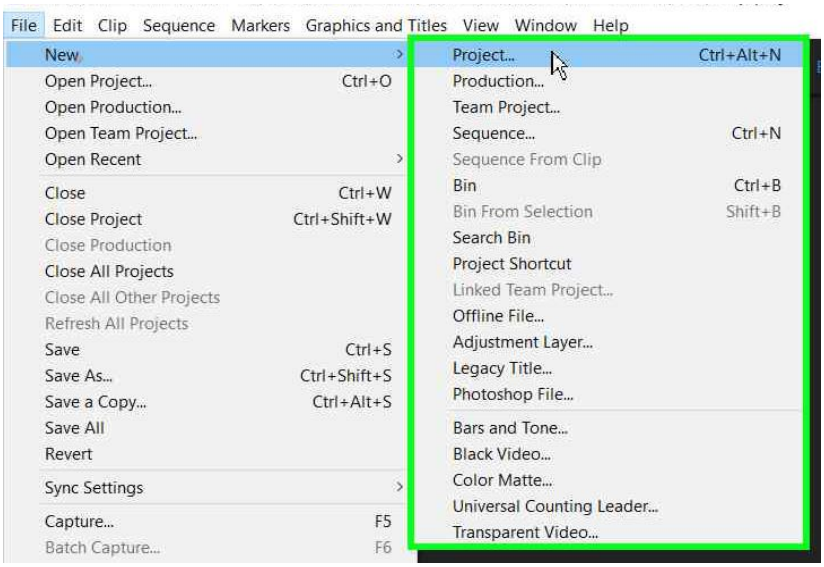

Windows BGI

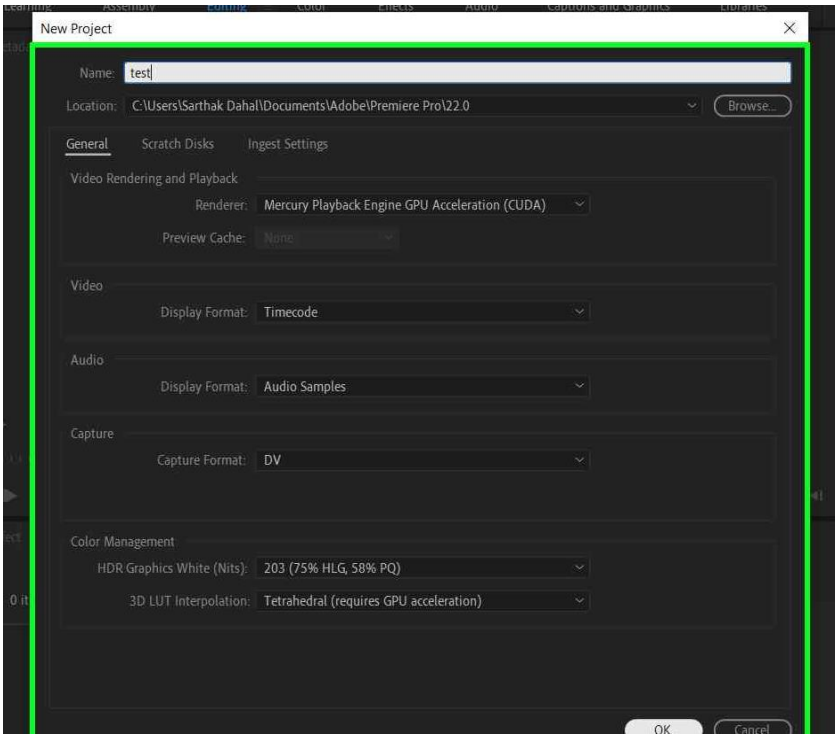
Lab No: 5

WRITE A PROGRAM/ ALGORITHM TO IMPORT, EDIT AND EXPORT A VIDEO ON SOME VIDEO AUTHORING SOFTWARE. E.g. Premiere Rush/Pro.

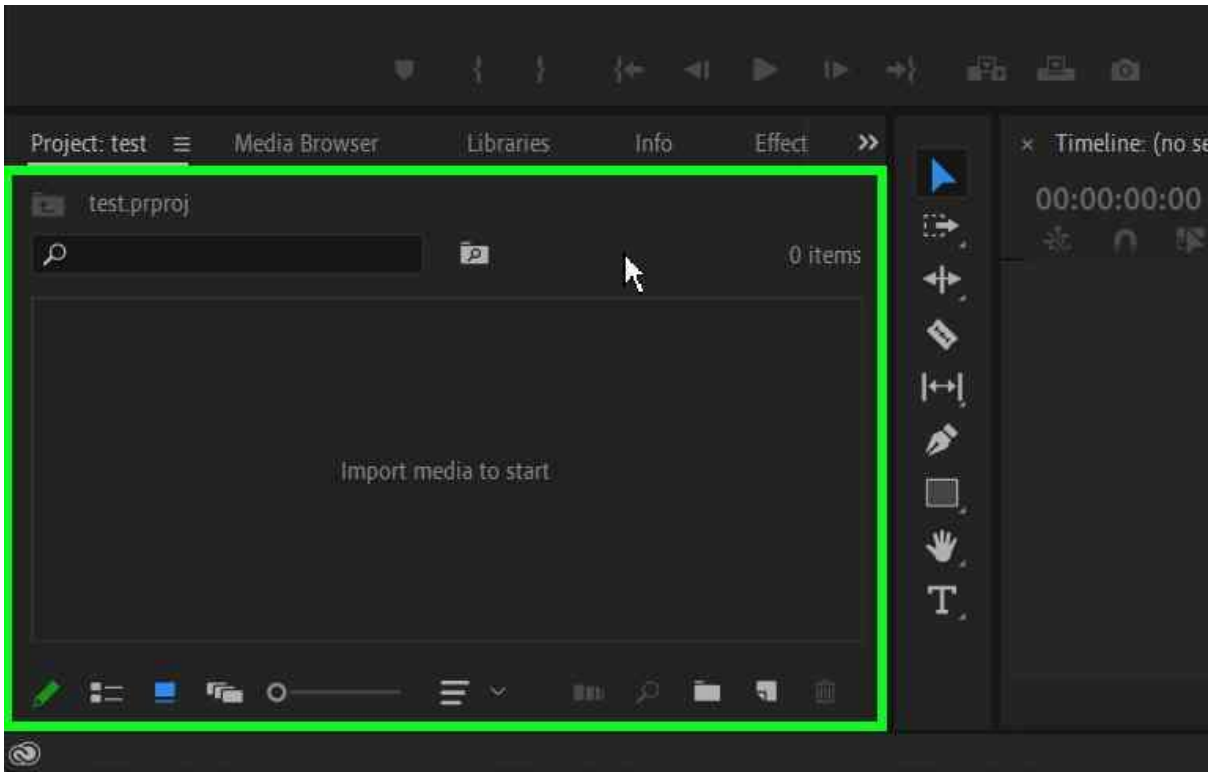
Steps for simple IMPORT, EDIT AND EXPORT of a video are given below:

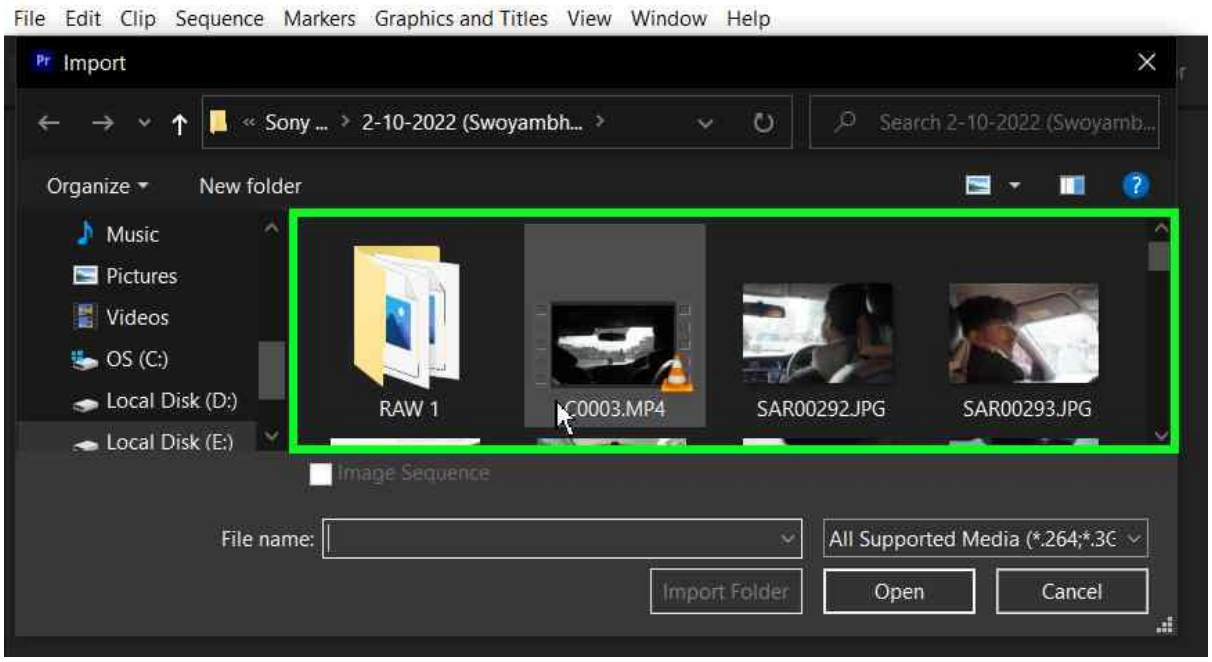
i. First open Adobe Premier Pro software and create a new project with the desired name by going to File -> New -> Project.



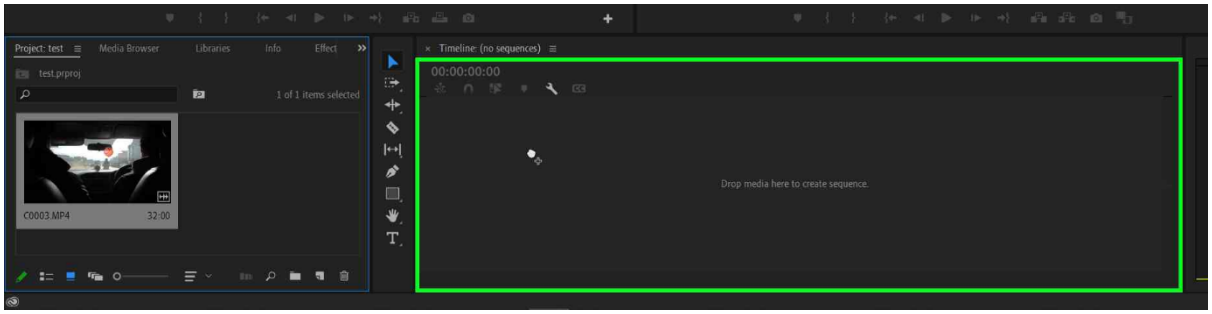


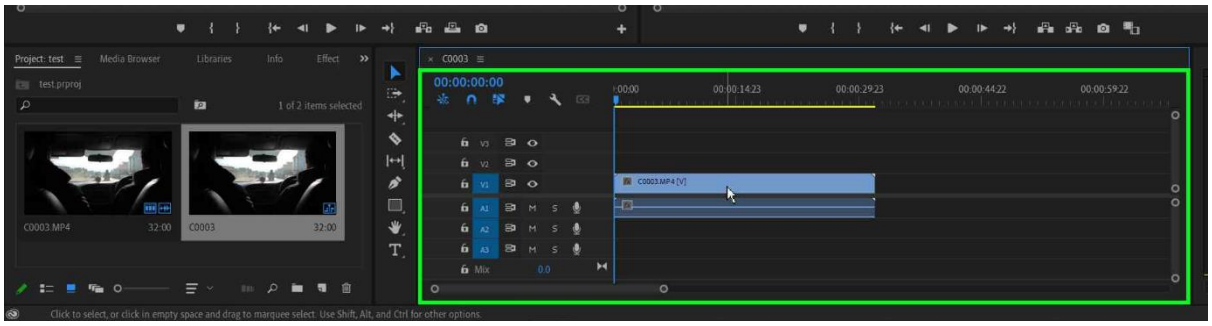
ii. Then click on File -> Import (or Import Media). This is where you'll be taken to choose the video clips you want to edit. Once done this, the clips will be placed in Media Browser or Media Pool.

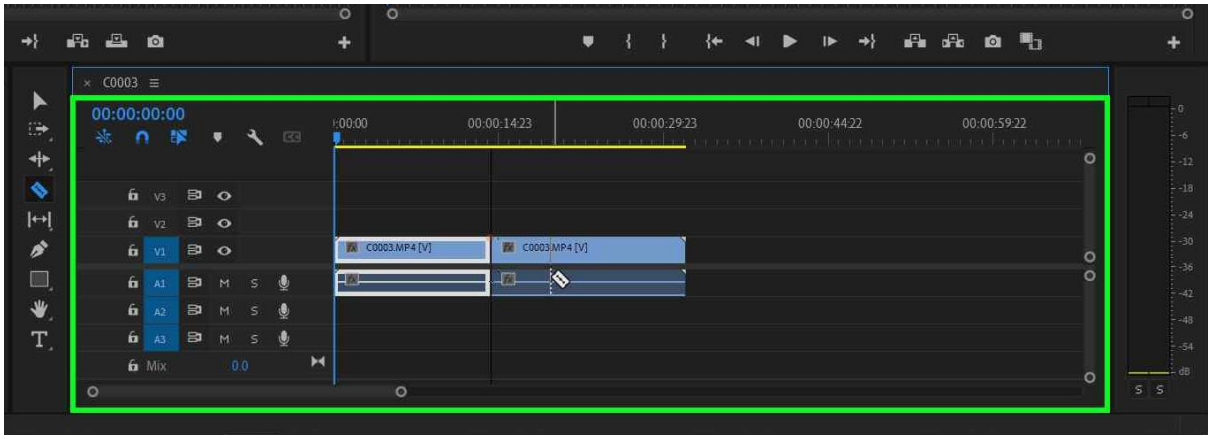




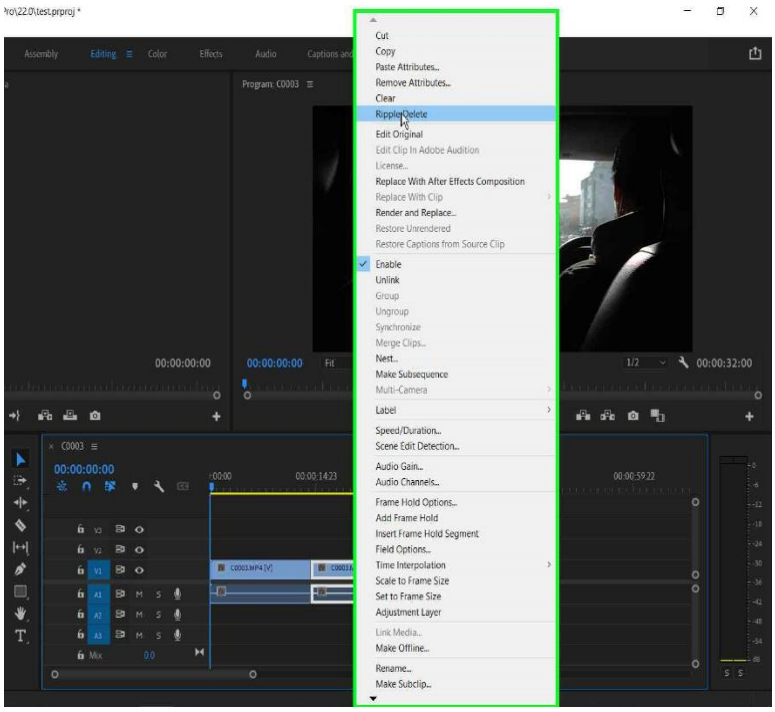
iii. Bring the clips in the timeline by dragging the video to the timeline section of the software. Now to slice the video into two different parts, select the cut tool by pressing C on your keyboard (For Windows Users) and then click at the portion of the video you want to slice.



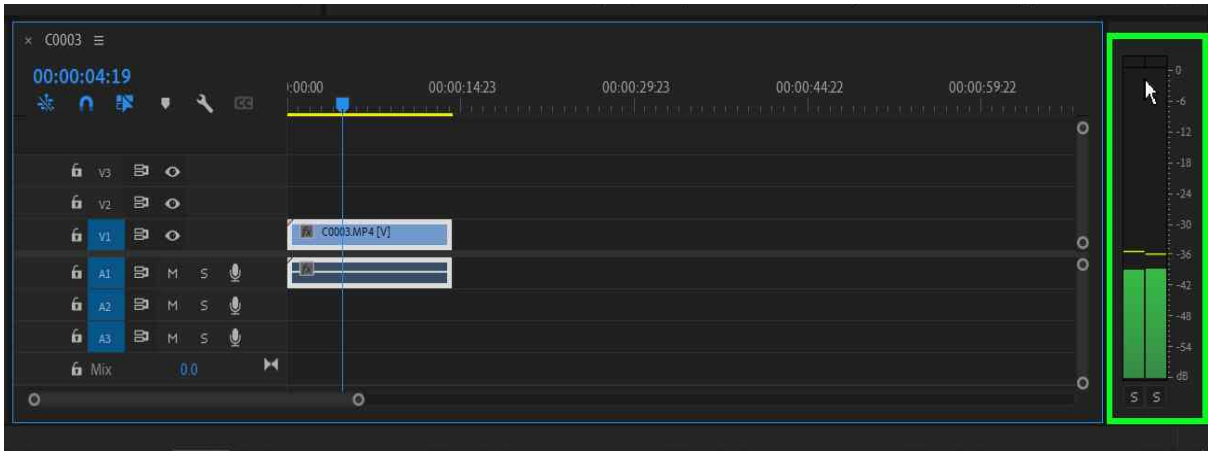




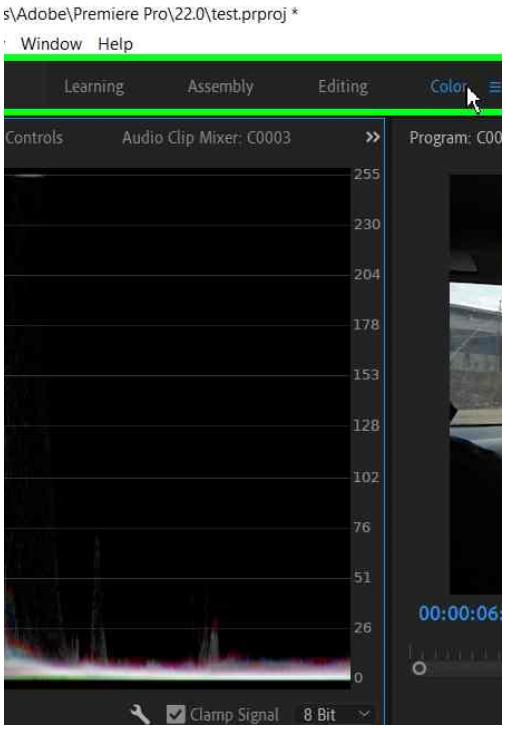
iv. To delete any clips in the video, right click on the portion and select ripple delete.

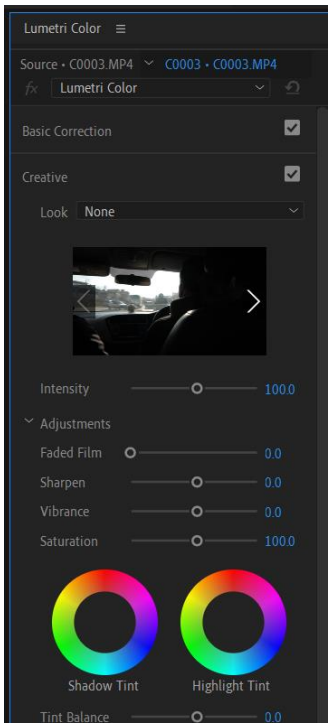


v. To add the audio to the video simply place the clip under your video file on your timeline and match up the video to the audio. If you don't want to do this manually, most programs will offer a way to synchronize the two by simply right-clicking on both files and hitting Synchronize.

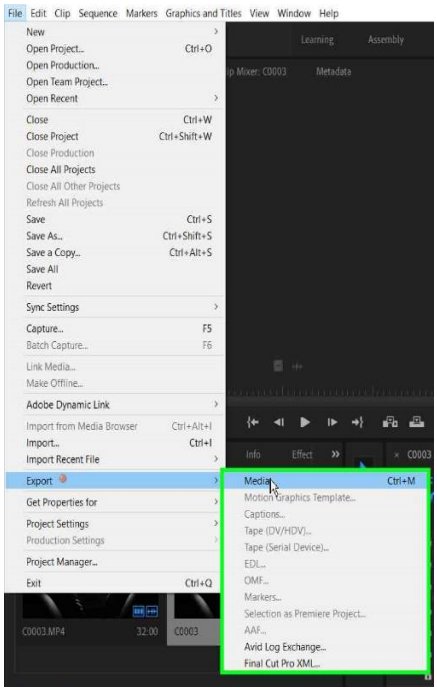


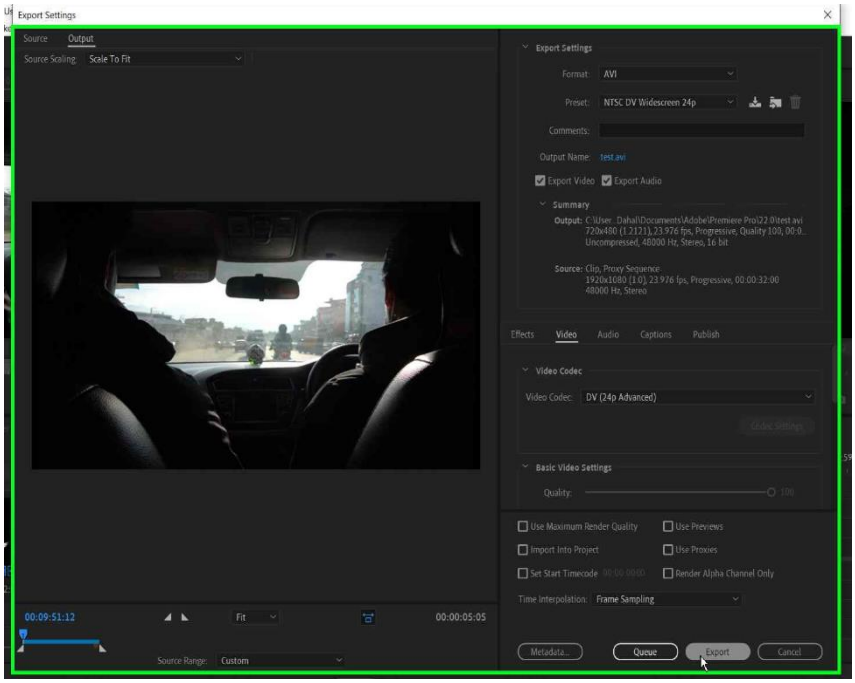
vi. There are some simple tools you can use to correct your footage. These tools live in the Color section of the editing program. With Premiere Pro, it's called Lumetri Color. These tools will allow you to raise or lower the Contrast, brighten, or darken the Exposure, change the Color Temperature, and raise or lower the Shadows and Highlights etc.





vii. Now the video is exported as a particular type of file, so that it can be uploaded or played in a specific way. It is also decided to export the video to whatever folder or destination you want the video to live. Video is generally exported at H.264, which will produce an MP4 file. You can export the video using command Ctrl+M or by clicking "File" then "Export as". The export setting can be changed as per the users' requirements.





viii) Now we can play the edited video in any video player as per our requirement.

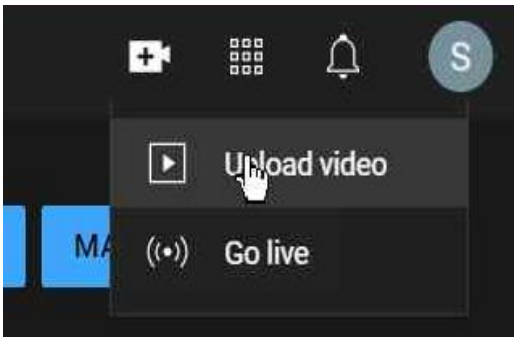
Lab No: 6

WRITE A STEP-BY-STEP PROGRAM/ ALGORITHM TO UPLOAD THE VIDEO TO YOUTUBE AND EMBED THAT VIDEO ON A HTML PAGE.

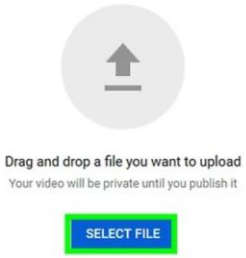
Step by step algorithm to upload video to YouTube:

i. Go to https://www.youtube.com in a web browser
Note: If you are not already signed into your YouTube account click SIGN IN at the top right corner of the page to sign in. Select your YouTube account and enter your password or use your email address to sign into your YouTube account.

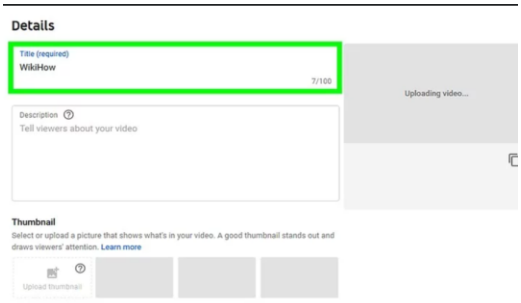
ii. Click the camera icon with a plus (+) sign and click Upload video. It's in the upper right corner. This takes you to the YouTube Studio web page with an "Upload Video" window in the center of the page.



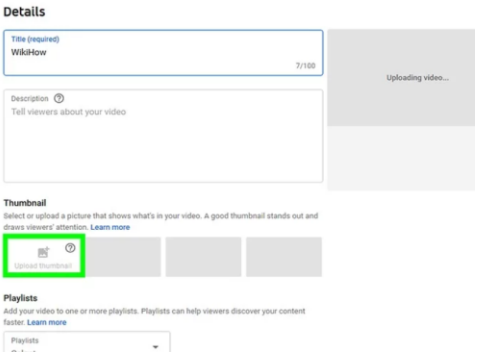
iii. Drag your video file to the arrow on the window. Alternatively, you can click the blue "SELECT FILE" button, browse to the file on your computer, and then click open.



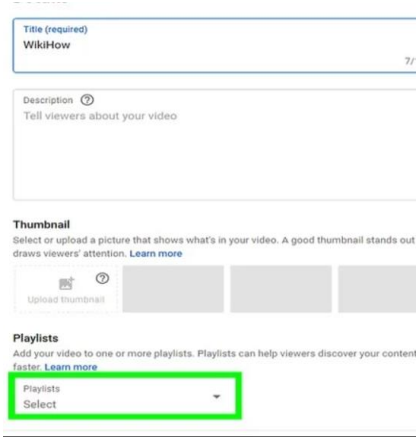
iv. Add a video title and description. The title is required, the description is optional. The title can be up to 100 characters long. Use large box labeled "Description" to add a description to the video. Info in the description will appear below your video while viewing.



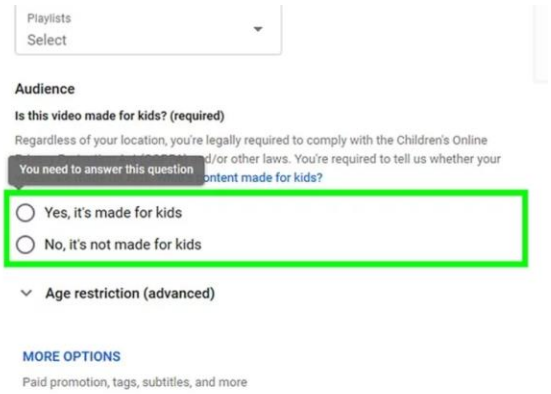
v. Click a thumbnail you want to use. The video's thumbnail is a still photo that represents the video on your channel and in search results.



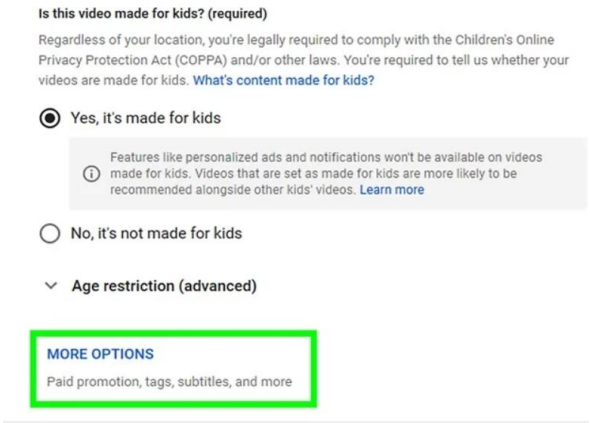
vi. Add the video to a playlist (Optional). If you want to add your video to a playlist, click the "playlist" drop down menu and select a playlist you have created to add your video to.



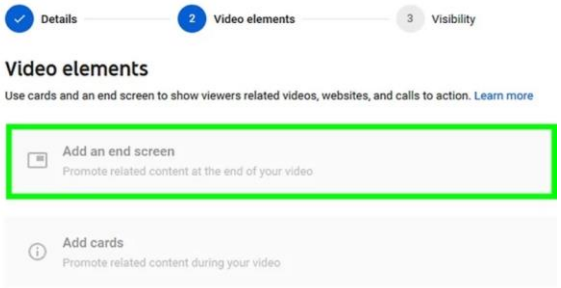
vii. Select if the video is made for kids or not. If your video is made for kids, click the radio button next to "Yes, it's made for kids". If your video is not made for kids, click the radio button next to "No, it's not made for kids".



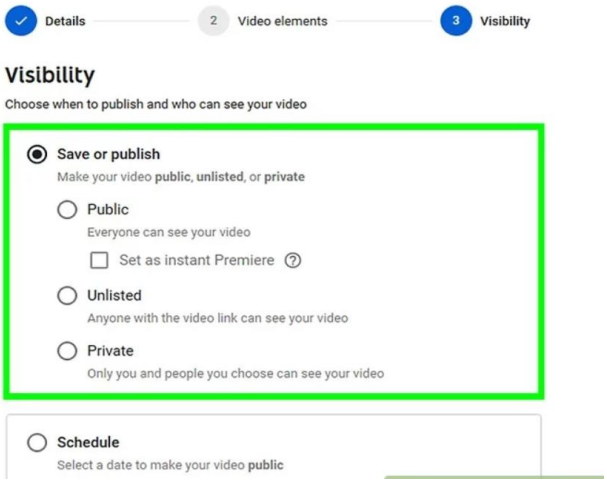
viii. Click “More Options” and click” Next” when you are finished. The additional options include Paid Promotions, Tags, Language, Subtitle and Closed captions, Recording Date and Location, License and distribution, Category, Comments, and ratings.



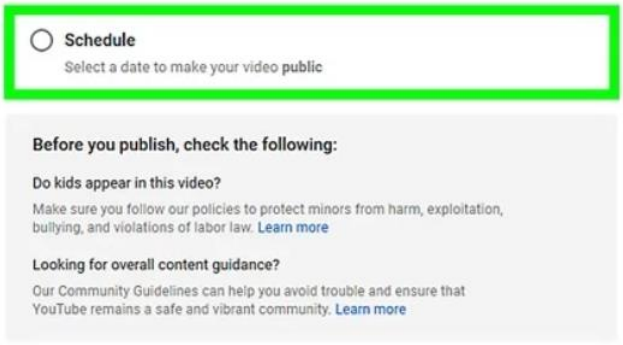
ix. Add an End Screen or cards to your video (Optional) and click “Next”. An end screen is a screen that appears at the end of the video to promote related material on your channel. Cards allow you to promote your material during the video.



x. Select a visibility level. There are three visibility option (Public, Unlisted, Private).Click the radio button next to the option you prefer.



xi. Schedule a date to go public (Optional). If you do not want the video to go public right away, you can schedule a time to go public. To schedule a time, click the radio button next to “Schedule”. Then click the drop-down menu with the date and select the date you want it to get public.

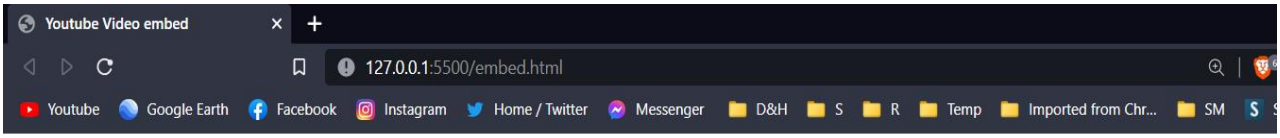


xii. Click the blue “save” or “Schedule” button. It’s at the bottom-right corner of the window. This will publish the video to your YouTube channel at the scheduled date and time, or after the video is finished processing.

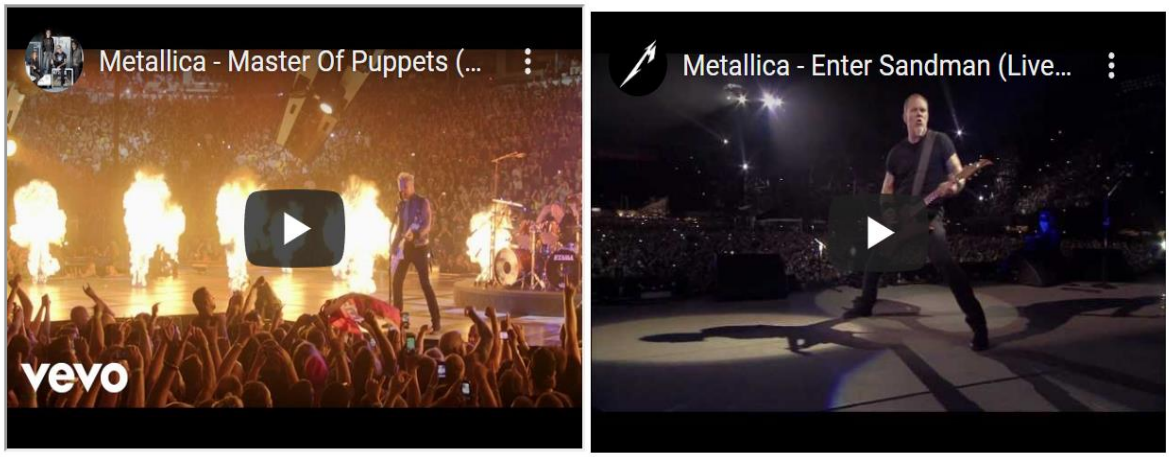
HTML code to embed the video on a HTML page:

```
<!DOCTYPE html>
<html>
<head>
  <title>Youtube Video embed</title>
</head>
<body>
  <h2> Youtube Video Embed Example </h2>
  <iframe width="390" height="270" src="https://www.youtube.com/embed/kV-2Q8QtCY4" frameborder="1"    allowfullscreen>
  </iframe>
  <iframe width="390" height="270" src="https://www.youtube.com/embed/87by1DjfxLw" title="YouTube video player" frameborder="0" allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe>
</body>
</html>
```

Output:



Youtube Video Embed Example



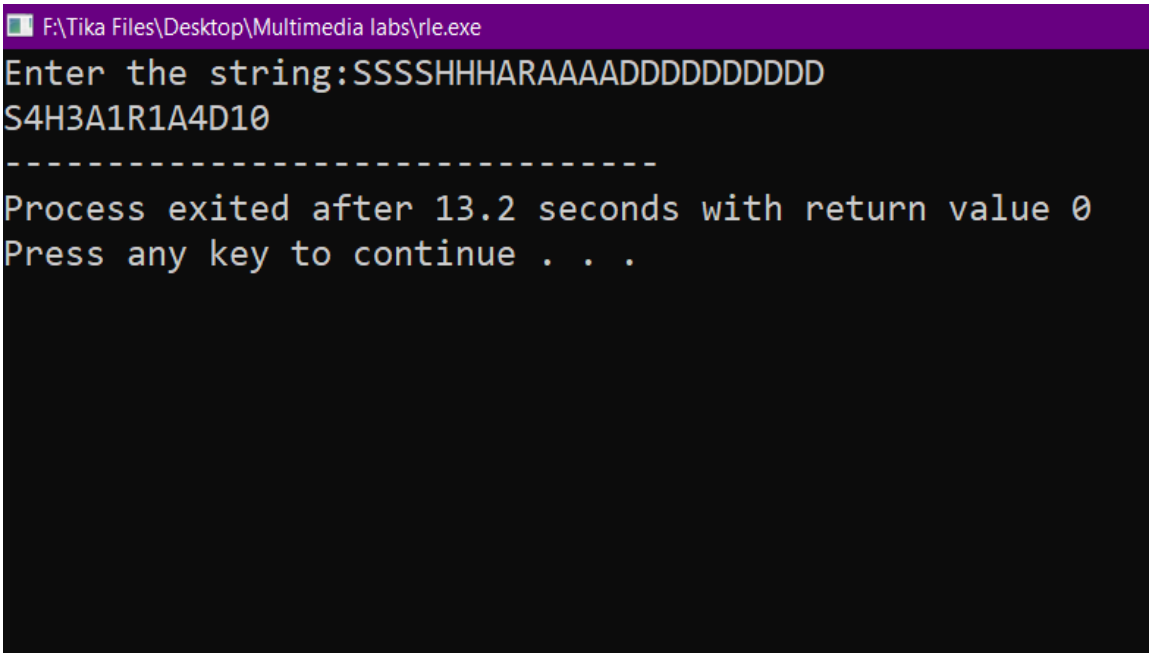
Lab No: 7

WRITE A PROGRAM TO IMPLEMENT RUN LENGTH CODING ON GIVEN SET OF STRINGS.

Source Code:

```
#include <iostream>
using namespace std;
void printRLE (string str)
{
   int n = str.length() ;
   for (int i=0;i<n;i++){
   // Count occurrences of current character
          int count= 1;
          while (i < n-1 && str[i] == str[i+1]){
          count++;
          i++;
          }
          // Print character and its count
          cout<<str[i] <<count;
     }
}
int main()
          {
          string str;
          cout<<"Enter the string:";
          cin>>str;
          printRLE(str);
          return 0;
    }
```

Output:


F:\Tika Files\Desktop\Multimedia labs\rle.exe
Enter the string:SSSSHHHARAAAADDDDDDDDDD
S4H3A1R1A4D10

Process exited after 13.2 seconds with return value 0
Press any key to continue . . .

Lab No: 8

WAP TO IMPLEMENT HUFFMAN CODING ON SOME GIVEN CHARACTERS AND THEIR FREQUENCIES.

Source Code:

```
#include<string.h>
#include<stdio.h>
#include<stdlib.h>
typedef struct node
{
	char ch;
	int freq;
	struct node *left;
	struct node *right;
}node;
node * heap[100];
int heapSize=0;
void Insert(node * element){
	heapSize++;
	heap[heapSize] = element;
	int now = heapSize;
	while(heap[now/2] -> freq > element -> freq){
		heap[now] = heap[now/2];
		now /= 2;
	}
	heap[now] = element;
}
node * DeleteMin(){
	node * minElement,*lastElement;
	int child,now;
	minElement = heap[1];
	lastElement = heap[heapSize--];
	for(now = 1; now*2 <= heapSize ;now = child) {
	    child = now*2;
	      if(child != heapSize && heap[child+1]->freq < heap[child] -> freq ) {
	          child++;
	      }
	      if(lastElement -> freq > heap[child] -> freq) {
```

```
                heap[now] = heap[child];
            }
            else{
                break;
            }
        }
        heap[now] = lastElement;
        return minElement;
}
void print(node *temp,char *code){
        if(temp->left==NULL && temp->right==NULL) {
            printf("char %c code %s\n",temp->ch,code);
            return;
        }
        int length = strlen(code);
        char leftcode[10],rightcode[10];
        strcpy(leftcode,code);
        strcpy(rightcode,code);
        leftcode[length] = '0';
        leftcode[length+1] = '\0';
        rightcode[length] = '1';
        rightcode[length+1] = '\0';
        print(temp->left,leftcode);
        print(temp->right,rightcode);
}
int main(){
       heap[0] = (node *)malloc(sizeof(node));
        heap[0]->freq = 0;
        int n ;
        printf("Enter the no of characters: ");
        scanf("%d",&n);
        printf("Enter the characters and their frequencies:\n ");
        char ch;
        int freq,i;
        for(i=0;i<n;i++){
            scanf(" %c",&ch);
            scanf("%d",&freq);
            node * temp = (node *) malloc(sizeof(node));
            temp -> ch = ch;
            temp -> freq = freq;
            temp -> left = temp -> right = NULL;
```

```
        Insert(temp);
    }
    if(n==1) {
        printf("char %c code 0\n",ch);
        return 0;
    }
    for(i=0;i<n-1 ;i++) {
        node * left = DeleteMin();
        node * right = DeleteMin();
        node * temp = (node *) malloc(sizeof(node));
        temp -> ch = 0;
        temp -> left = left;
        temp -> right = right;
        temp -> freq = left->freq + right -> freq;
        Insert(temp); }
    node *tree = DeleteMin();
    char code[10];
    code[0] = '\0';
    print(tree,code);
}
```

Output: