**DSP LAB**

**Assignment – 2**

Question - 1 : Write a Program that ask user to enter the string (Note : string can contain only ‘(’ , ’)’ , ’[’ , ’]’, ‘{’ , ‘}’ these mentioned 6 characters only ) You have to determine that input string is valid or not.

Source code:

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

int main()

{

char arr[50];

int i,p=0,pc=0,b=0,bc=0,sb=0,sbc=0,indp=0,indpc=0,indb=0,indbc=0,indsb=0,indsbc=0;

printf("Enter string:");

scanf("%s",arr);

for(i=0;i<strlen(arr);i++)

{

if(arr[i]=='(')

{

p++;

indp=i;

}

else if((arr[i]==')') && (i!=0) && p!=0)

{

pc++;

indpc=i;

if((indp+2==indpc)&&(arr[indpc-1]!=')'))

{

printf("false");

exit(0);

}

}

else if(arr[i]=='{')

{

b++;

indb=i;

}

else if((arr[i]=='}') && (i!=0) && b!=0)

{

bc++;

indbc=i;

if((indb+2==indbc)&&(arr[indbc-1]!='}'))

{

printf("false");

exit(0);

}

}

else if(arr[i]=='[')

{

sb++;

indsb=i;

}

else if((arr[i]==']') && (i!=0) &&sb!=0)

{

sbc++;

indsbc=i;

if((indsb+2==indsbc)&&(arr[indsbc-1]!=']'))

{

printf("false");

exit(0);

}

}

else{

printf("false");

exit(0);

}

}

if((p==pc)&&(b==bc)&&(sb==sbc))

{

printf("true");

}

else{

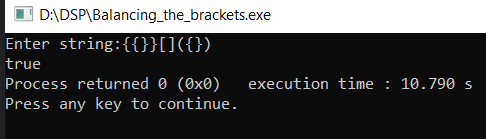
printf("false");

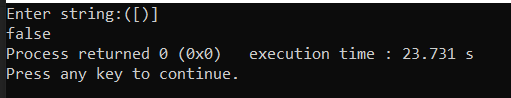
}

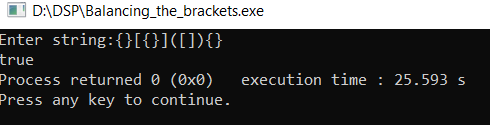
return 0;

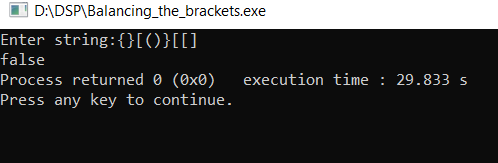
}

Output:









2. Create a singly link list which contains either 0 or 1 as element only

Source Code:

#include<stdio.h>

#include<stdlib.h>

struct node

{

int data;

struct node \*next;

}\*p;

struct node \*n;

//Function to Add the items at beginning of the List

void addbeg(int num)

{

struct node \*temp;

temp=(struct node \*)malloc(sizeof(struct node));

temp->data=num;

if(p==NULL)

{

p=temp;

p->next=NULL;

}

else

{

temp->next=p;

p=temp;

}

}

//Function to Add the items at end of the List

void addend(int num)

{

struct node \*temp,\*r;

temp=(struct node \*)malloc(sizeof(struct node));

temp->data=num;

r=(struct node \*)p;

if(p==NULL)

{

p=temp;

p->next=NULL;

}

else

{

while(r->next!=NULL)

r=r->next;

r->next=temp;

r=temp;

r->next=NULL;

}

}

int count()

{

struct node \*n;

int c=0;

n=p;

while(n!=NULL)

{

n=n->next;

c++;

}

return(c);

}

//Function to Display the items of the List

void display(struct node \*r)

{

r=p;

if(r==NULL)

{

printf("No element in the list");

return;

}

while(r!=NULL)

{

printf("->%d",r->data);

r=r->next;

}

printf("\n");

}

//Function to calculate decimal value from binary values

void Calculate\_Decimal\_value(struct node \*r)

{

int num;

r=p;

if(r==NULL)

{

printf("No element in the list");

return;

}

int btod = 0,bnum;

int c = count();

//printf("c:%d",c);

int i=c-1;

while(r!=NULL && i>=0)

{

bnum = r->data;

btod = btod + bnum\*(pow(2,i));

printf("->%d",r->data);

r=r->next;

--i;

}

printf("\nDecimal value for given binary number :%d",btod);

}

void main()

{

int i;

p=NULL;

while(1)

{

printf("\n1.Insert a Binary number at beginning");

printf("\n2.Insert a Binary number at last");

printf("\n3.Display the Binary numbers");

printf("\n4.Calculate Decimal value and Exit from program");

printf("\n\nEnter your choice : ");

scanf("%d",&i);

if(i == 1)

{

int num;

printf("please enter number:");

scanf("%d",&num);

if(num==1 || num==0)

{

addbeg(num);

}

else{

printf("Please enter 0 or 1");

}

}

else if(i == 2)

{

int num;

printf("please enter number:");

scanf("%d",&num);

if(num==1 || num==0)

{

addend(num);

}

else{

printf("Please enter 0 or 1");

}

}

else if(i==3)

{

printf("The elements in list are :\n ");

display(n);

}

else if (i==4){

Calculate\_Decimal\_value(n);

exit(0);

}

else{

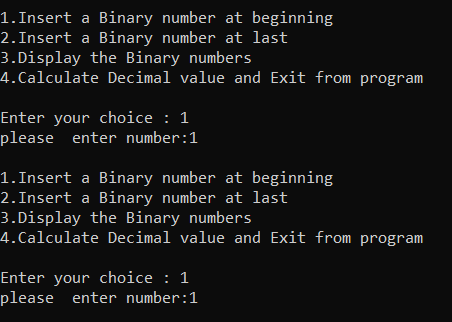
printf("please enter valid choice:");

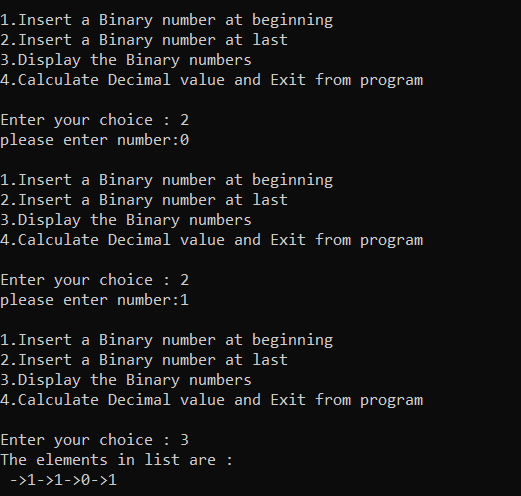
}

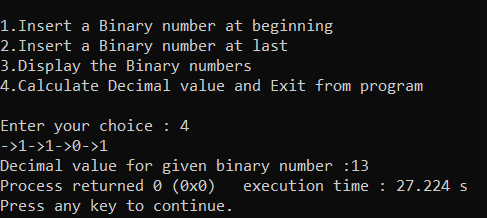
}

}

Output:







3. Write a source code to create a singly circular linked list. A node should consist of • Character array to store three strings in “info” part • Address of next node as “link” part.

Source Code:

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<math.h>

struct node {

char scl[3][10];

struct node \*next;

};

struct node \*head = NULL;

void add()

{

struct node \*newnode,\*temp;

newnode = (struct node \*)malloc(sizeof(struct node));

if(newnode == NULL)

{

printf("\nOverflow\n");

}

else

{

if(head == NULL)

{

head = newnode;

newnode -> next = head;

for(int i=0;i<3;i++)

{

scanf("%s",newnode -> scl[i]);

}

}

else

{

temp = head;

while(temp -> next != head)

{

temp = temp -> next;

}

temp -> next = newnode;

newnode -> next = head;

for(int i=0;i<3;i++)

{

scanf(" %s",newnode->scl[i]);

}

}

printf("\n Node is inserted to the List\n");

}

}

void display()

{

struct node \*p;

p =head;

printf("\n Colors are \n");

while(p->next != head)

{

for(int i=0;i<3;i++)

{

printf(" %s\t", p->scl[i]);

}

p = p->next;

printf("\n");

}

int j=0;

while(j<3)

{

printf(" %s\t ", p->scl[j]);

j++;

}

printf("\n");

}

void Displaysecond\_string()

{

struct node \*p;

p =head;

while(p->next != head)

{

printf(" %s \n", p->scl[1]);

p = p->next;

}

printf(" %s \n", p->scl[1]);

}

void Display\_colorReplace()

{

struct node \*p;

p =head;

strcpy(p->scl[0],"color");

while(p->next != head)

{

p = p->next;

}

strcpy(p->scl[0],"color");

display();

}

int isMatch(){

struct node \*p;

p =head;

while(p->next != head)

{

if(strcmp(p->scl[2],p->next->scl[0])!=0)

{

return 0;

}

p = p->next;

}

return 1;

}

int main()

{

int nodes;

printf("Enter number of Nodes :");

scanf("%d",&nodes);

printf("Enter elements to Insert in to singly circular linked list: \n");

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

{

add();

printf("Please Enter Elements for next node:\n");

}

display();

printf("\nSecond string colors in the Linked list are: \n");

Displaysecond\_string();

if(isMatch())

{

printf("\n String Matching");

}

else

{

printf("\n String not Matching ");

}

Display\_colorReplace();

if(isMatch())

{

printf("\n String Matching ");

}

else

{

printf("\n String not Matching ");

}

return 0;

}

Output:

