CC-LAB RECORD

Exercise -1

Aim: Write a Python program to print a specified list after removing the 0th, 4th and 5th elements.

Program:

```
li = [1, 2, 3, 4, 5, 6,'srkr',9.5,'21-5-222']
del li[0]
del li[4:6]
print(li)
```

output:

[2, 3, 4, 5, 9.5, '21-5-222']

Exercise-2 :

Aim: Write a Python program to generate a 3*4*6 3D array whose each element is *.

Program:

```
import numpy as np
li = ["*"] * 72
arr = np.array(li)
arr = arr.reshape((3, 4, 6))
print(arr)
```

Output:

```
[[['*' '*' '*' '*' '*' '*']
  ['*' '*' '*' '*' '*' '*']
  ['*' '*' '*' '*' '*' '*']
  ['*' '*' '*' '*' '*' '*']]

 [['*' '*' '*' '*' '*' '*']
  ['*' '*' '*' '*' '*' '*']
  ['*' '*' '*' '*' '*' '*']
  ['*' '*' '*' '*' '*' '*']]

 [['*' '*' '*' '*' '*' '*']
  ['*' '*' '*' '*' '*' '*']
```

```
['*' '*' '*' '*' '*' '*']
['*' '*' '*' '*' '*' '*']]]
```

Exercise-3 :

Aim: Write a Python program to generate all permutations of a list in Python

Program:

```
from itertools import permutations
l = list(permutations(['A','B','C']))
print(l)
```

Output:

```
[('A', 'B', 'C'), ('A', 'C', 'B'), ('B', 'A', 'C'), ('B', 'C', 'A'), ('C', 'A', 'B'), ('C', 'B', 'A')]
```

Exercise -4:

Aim: Write a Python program to check whether a list contains a sub list .

Program:

```
def sub_lists(l):
    lists = []
    for i in range(len(l) + 1):
        for j in range(i):
            lists.append(l[j:i])
    return lists
li = [1, 2, 3, 4, 5]
sub_list = [1, 2]
print(sub_list in sub_lists(li))
```

output:
True

Exercise -5:

Aim: Write a Python program to change the position of every n-th value with the (n+1)th in a list.

Program:

```
li = [1, 2, 3, 4, 5]
for i in range(len(li) - 1):
    li[i] = li[i] + li[i + 1]
    li[i + 1] = li[i] - li[i + 1]
    li[i] = li[i] - li[i + 1]
print(li)
```

Output:

[2, 3, 4, 5, 1]

Exercise-6:

Aim: Write a Python program to sort a list of nested dictionaries.

Program:

```
my_list = [{'key': {'subkey': 1}}, {'key': {'subkey': 10}}, {'key': {'subkey': 5}}]
print("Original List: ")
print(my_list)

my_list.sort(key=lambda e: e['key']['subkey'], reverse=True)
print("Sorted List: ")
print(my_list)
```

Output:

```
Original List:
[{'key': {'subkey': 1}}, {'key': {'subkey': 10}}, {'key': {'subkey': 5}}]
Sorted List:
[{'key': {'subkey': 10}}, {'key': {'subkey': 5}}, {'key': {'subkey': 1}}]
```

Exercise-7

Aim: Write a Python program to move all zero digits to end of a given list of numbers.

Expected output:

Original list:

[3, 4, 0, 0, 0, 6, 2, 0, 6, 7, 6, 0, 0, 0, 9, 10, 7, 4, 4, 5, 3, 0, 0, 2, 9, 7, 1]

Move all zero digits to end of the said list of numbers:

[3, 4, 6, 2, 6, 7, 6, 9, 10, 7, 4, 4, 5, 3, 2, 9, 7, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0]

Program:

```
original_list = [3, 4, 0, 0, 0, 6, 2, 0, 6, 7, 6, 0, 0, 0, 9, 10, 7, 4, 4, 5, 3, 0, 0, 2, 9, 7,
1]
non_zeroes = []
zeroes = []
for i in range(len(original_list)):
    if original_list[i] == 0:
        zeroes.append(0)
    else:
        non_zeroes.append(original_list[i])
non_zeroes.extend(zeroes)
print(non_zeroes)
```

Output:

[3, 4, 6, 2, 6, 7, 6, 9, 10, 7, 4, 4, 5, 3, 2, 9, 7, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0]

Exercise -8:

Aim: Write a Python program to read a matrix from console and print the sum for each column. Accept matrix rows, columns and elements for each column separated with a space(for every row) as input from the user.

Program :

```
rows = int(input())
columns = int(input())
matrix = []
for i in range(rows):
    matrix.append(list(map(int, input().split())))
sums = []
```

```
for j in range(columns):
    sum = 0
    for i in range(rows):
        sum += matrix[i][j]
    sums.append(sum)
print(sums)
```

Output:

1 2 3
4 5 6
7 8 9
[12, 15, 18]

Exercise -9:

Aim: Write a Python program to read a square matrix from console and print the sum of matrix primary diagonal. Accept the size of the square matrix and elements for each column separated with a space (for every row) as input from the user.

Program:

```
size = int(input())
matrix = []
sum_of_primary_diagonal = 0
for i in range(size):
    matrix.append(list(map(int, input().split())))
for i in range(size):
    sum_of_primary_diagonal += matrix[i][i]
print(sum_of_primary_diagonal)
```

Output:

1 2 3
4 5 6
7 8 9

15

Exercise -10:

Aim: Write a Python function that takes a list of words and return the longest word and the length of the longest one .

Program:

```
list_of_words = list(map(str, input().split()))
max_len = 0
max_word = ""
for i in list_of_words:
    if len(i) > max_len:
        max_word = i
        max_len = len(i)
print(max_word, max_len)
```

Output:

```
MSDhoni is the best capt
MSDhoni 7
```

Exercise -11

Aim: Write a Python program to remove the n^{th} index character from a nonempty string

Program :

```
string = "Hello! How are you?"
string = list(string)
string.pop(5)
print("".join(string))
```

Output:

```
Hello How are you?
```

Exercise -12:

Aim: Write a Python program to remove the characters which have odd index values of a given string.

Program:

```
def odd_values_string(str):
  result = ""
  for i in range(len(str)):
    if i % 2 == 0:
      result = result + str[i]
  return result
print(odd_values_string('srkr'))
print(odd_values_string('college'))
```

Output:

```
sk
clee
```

Exercise -13

Aim: Write a Python program that accepts a comma separated sequence of words as input and prints the unique words in sorted form (alphanumerically).

Program:

```
items = input("Input comma separated sequence of words")
words = [word for word in items.split(",")]
print(",".join(sorted(list(set(words)))))
```

Output:

```
Input comma separated sequence of wordssrkr,college,srkr
college,srkr
```

Exercise -14

Aim: Write a Python function to get a string made of 4 copies of the last two characters of a specified string (length must be at least 2).

Program:

```
def insert_end(str):
        sub_str = str[-2:]
        return sub_str * 4

print(insert_end('srkr'))
print(insert_end('college'))
```

output:

krkrkrkr
gegegege

Exercise -15

Aim: Write a Python program to sort a string lexicographically

Program:

```
def sortStringInLexo(string):
      words = string.split()
      words.sort()
      for word in words:
            print( word ) sortStringInLexo ("srkr college Srkr")
```

Output: Srkr college

Output: srkr

Exercise :16

Aim: Implement singly linked list(insertion, deletion, display)

Program:

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
        int info;
        struct node *next;
};
typedefstruct node *nodeptr;
nodeptrgetnode();
void create();
void insert();
void del();
void display();
nodeptr list;
intmain()
{
        intch;
        clrscr();
        list=getnode();
        while(1)
        {
                printf("\n***************\nMENU\n************\n");
                printf("\n1.create\n2.insert\n3.delete\n4.display\n5.exit\n");
                printf("\nenter your choice:");
                scanf("%d",&ch);
                switch(ch)
                {
                        case 1:         create();
                                display();
                                break;
                        case 2:         insert();
                                display();
```

```
                        break;
                case 3:     del();
                        display();
                        break;
                case 4:     display();
                        break;
                case 5:     exit(0);
                }
        }
}
nodeptrgetnode()
{
        nodeptr p;
        p=(nodeptr)malloc(sizeof(struct node));
        p->info=0;
        p->next=NULL;
        return p;
}
void create()
{

nodeptr p1,p2;
    p1=list;
    p2=getnode();
printf("\nEnter the at end -999\n");
printf("\nEnter the number:");
scanf("%d",&p2->info);
    while(p2->info!=-999)
    {
            p1->next=p2;
            p1=p2;
            p2=getnode();
            printf("\nEnter the number:");
            scanf("%d",&p2->info);
    }
      list=list->next;
}
void display()
{
        nodeptr p;
```

```
        p=list;
        printf("\nelements are:");
        while(p->next!=NULL)
        {
                printf("%d-->",p->info);
                p=p->next;
        }
}
void insert()
{
        nodeptrp,p1;
        intch,pos,i;
        p=getnode();
        p1=list;
        printf("\nenter the insert number:");
        scanf("%d",&p->info);
        printf("\n1.begining\n2.given position\n");
        printf("enter your choice:");
        scanf("%d",&ch);
        switch(ch)
        {
                case 1:         p->next=list;
                        list=p;
                        break;
                case 2:         printf("\nEnter the position to insert:");
                        scanf("%d",&pos);
                        for(i=1;i<pos-1;i++)
                                p1=p1->next;
                        p->next=p1->next;
                        p1->next=p;
                        break;
                default:printf("\nwrong choice");
        }
}
void del()
{
        nodeptr p;
        intch,i,pos;
        p=list;
        printf("\n1.delete at begining\n2.delete at given pos\n");
```

```
        printf("\nenter your choice:");
        scanf("%d",&ch);
        switch(ch)
        {
                case 1: list=list->next;
                        p->next=NULL;
                        break;
                case 2:         printf("\nEnter the position for delete:");
                        scanf("%d",&pos);
                        for(i=1;i<pos-1;i++)
                                p=p->next;
                        p->next=p->next->next;
                        break;
                default:printf("\nwrong choice");
        }
}
```

OutPut :

```
**********
MENU
**********
1.create
2.insert
3.delete
4.display
5.exit
enter your choice:2
enter the insert number:34
1.begining
2.given position
enter your choice:1
elements are:34-->
**********
MENU
**********
1.create
```

2.insert
3.delete
4.display
5.exit
enter your choice:2
enter the insert number:55
1.begining
2.given position
enter your choice:1
elements are:55-->34-->
****
MENU
****
1.create
2.insert
3.delete
4.display
5.exit
enter your choice:2
enter the insert number:12
1.begining
2.given position
enter your choice:2
Enter the position to insert:2
elements are:55-->12-->34-->
****
MENU
****
1.create
2.insert
3.delete
4.display
5.exit
enter your choice:4
elements are:55-->12-->34-->

****
MENU
****
1.create
2.insert
3.delete
4.display
5.exit

enter your choice:3
1.delete at begining
2.delete at given pos
enter your choice:1
elements are:12-->34-->
****
MENU
****
1.create
2.insert
3.delete
4.display
5.exit
enter your choice:5

Exercise: 17

Aim: Implement doubly linked list(insertion, deletion, display)

Program:

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
        int info;
        struct node *pre,*next;
};
typedefstruct node *nodeptr;
```

```
nodeptrfirst,last;
void create();
nodeptrgetnode();
void insert();
void del();
void display();
intmain()
{
	intch;
	while(1)
	{
		printf("\n*****************\n\tMENU\n*************\n");
		printf("\n1.create\n2.insert\n3.delete\n4.display\n5.exit\n");
		printf("Enter your choice:");
		scanf("%d",&ch);
		switch(ch)
		{
			case 1:	create();
				display();
				break;
			case 2:	insert();
				display();
				break;
			case 3:	del();
				display();
				break;
			case 4:	display();
				break;
			case 5:	exit(0);
		}
	}
}
nodeptrgetnode()
{
	nodeptr p;
	p=(nodeptr )malloc(sizeof(struct node));
	p->info=0;
	p->pre=NULL;
	p->next=NULL;
	return p;
```

```
}
void create()
{
        nodeptr p1,p2,p3;
        p1=getnode();
        p2=getnode();
        p3=p1;
        printf("\nEnter the number -999 at END");
        printf("\nEnter the number:");
        scanf("%d",&p2->info);
        while(p2->info!=-999)
        {
                p1->next=p2;
                p2->pre=p1;
                p1=p2;
                p2=getnode();
                printf("\nEnter the number:");
                scanf("%d",&p2->info);
        }
        p3=p3->next;
        p3->pre=NULL;
        first=p3;
        last=p1;
}
void display()
{
        nodeptr p1,p2;
        p1=first;
        p2=last;
        printf("\nElements in order:");
        while(p1!=NULL)
        {
                printf("%d-->",p1->info);
                p1=p1->next;
        }
        printf("\nElements in reverse order:");
        while(p2!=NULL)
        {
                printf("%d<--",p2->info);
                p2=p2->pre;
```

```
        }
}
void insert()
{
        intx,c,i;
        nodeptr p1,p2,p3;
        p1=getnode();
        printf("\nEnter the number for insert:");
        scanf("%d",&p1->info);
        printf("\n1.at begining\n2.at given position\n3.at end\n");
        printf("\nEnter your choice:");
        scanf("%d",&c);
        switch(c)
        {
                case 1:         p1->next=first;
                        first->pre=p1;
                        first=p1;
                        break;
                case 2:         printf("\nEnter the position for insert:");
                        scanf("%d",&x);
                        p2=first;
                        for(i=1;i<x-1;i++)
                                p2=p2->next;
                        p3=p2->next;
                        p2->next=p1;
                        p1->pre=p2;
                        p1->next=p3;
                        p3->pre=p1;
                        break;
                case 3:         last->next=p1;
                        p1->pre=last;
                        last=p1;
                        break;
        }
}

void del()
{
        intc,x,i;
        nodeptr p1,p2,p3;
```

```
printf("\nMENU FOR DELETION\n");
printf("\n1.beg\n2.given pos\n3.end\nenter your choice:");
scanf("%d",&c);
switch(c)
{
        case 1:         p1=first->next;
                        first->next=NULL;
                        p1->pre=NULL;
                        first=p1;
                        break;
        case 2:         printf("\nEnter the position for deletion:");
                        scanf("%d",&x);
                        p1=first;
                        for(i=1;i<x-1;i++)
                                p1=p1->next;
                        p2=p1->next;
                        p3=p2->next;
                        p1->next=p3;
                        p3->pre=p1;
                        p2->next=p2->pre=NULL;
                        break;
        case 3:         p1=last->pre;
                        p1->next=NULL;
                        last->pre=NULL;
                        last=p1;
                        break;
}}
```

Output:

```
******************
      MENU
************
1.create
2.insert
3.delete
4.display
5.exit
Enter your choice:1
```

```
Enter the number -999 at END
Enter the number:23-999

Enter the number:
Elements in order:23-->
Elements in reverse order:23<--
**********
      MENU
**********

1.create
2.insert
3.delete
4.display
5.exit
Enter your choice:2

Enter the number for insert:12

1.at begining
2.at given position
3.at end

Enter your choice:1

Elements in order:12-->23-->
Elements in reverse order:23<--12<--
**********
      MENU
**********

1.create
2.insert
3.delete
4.display
5.exit
Enter your choice:2

Enter the number for insert:45
```

1.at begining
2.at given position
3.at end

Enter your choice:2

Enter the position for insert:1

Elements in order:12-->45-->23-->
Elements in reverse order:23<--45<--12<--
****
MENU
****

1.create
2.insert
3.delete
4.display
5.exit
Enter your choice:3

MENU FOR DELETION

1.beg
2.given pos
3.end
enter your choice:3

Elements in order:12-->45-->
Elements in reverse order:45<--12<--
****
MENU
****

1.create
2.insert
3.delete
4.display
5.exit
Enter your choice:4

Elements in order:12-->45-->
Elements in reverse order:45<--12<--
****
MENU
****

1.create
2.insert
3.delete
4.display
5.exit
Enter your choice:5

Exercise: 18

Aim: Implement circular linked linkedlist(insertion, deletion, display)

Program :

```
#include<stdio.h>
struct node
{
        int info;
        struct node *next;
};
typedefstruct node *nodeptr;
nodeptrgetnode();
void create();
void insert();
void del();
void display();
nodeptr list;
main()
{
        intch;
        clrscr();
        list=getnode();
        list->info=-1;
        while(1)
        {
                printf("\n**********\nMENU\n**********\n");
                printf("\n1.create\n2.insert\n3.delete\n4.display\n5.exit\n");
                printf("\nenter your choice:");
                scanf("%d",&ch);
                switch(ch)
                {
                        case 1:         create();
                                        display();
                                        break;
                        case 2:         insert();
                                        display();
                                        break;
                        case 3:         del();
```

```
                        display();
                        break;
                case 4:     display();
                        break;
                case 5:     exit(0);
            }
        }
}
nodeptrgetnode()
{
        nodeptr p;
        p=(nodeptr)malloc(sizeof(struct node));
        p->info=0;
        p->next=NULL;
        return p;
}
void create()
{

nodeptr p1,p2;
    p1=list;
    p2=getnode();
printf("\nEnter the at end -999\n");
printf("\nEnter the number:");
scanf("%d",&p2->info);
    while(p2->info!=-999)
    {
            p1->next=p2;
            p1=p2;
            p2=getnode();
            printf("\nEnter the number:");
            scanf("%d",&p2->info);
    }
      p1->next=list;
}
void display()
{
        nodeptr p;
        p=list->next;
        printf("\nelements are:");
```

```
        while(p->info!=-1)
        {
                printf("%d-->",p->info);
                p=p->next;
        }
}
void insert()
{
        nodeptrp,p1;
        intch,pos,i;
        p=getnode();
        p1=list->next;
        printf("\nenter the insert number:");
        scanf("%d",&p->info);
        printf("\n1.begining\n2.given position\n");
        printf("enter your choice:");
        scanf("%d",&ch);
        switch(ch)
        {
                case 1:         p->next=list->next;
                        list->next=p;
                        break;
                case 2:         printf("\nEnter the position to insert:");
                        scanf("%d",&pos);
                        for(i=1;i<pos-1;i++)
                                p1=p1->next;
                        p->next=p1->next;
                        p1->next=p;
                        break;
                default:printf("\nwrong choice");
        }
}
void del()
{
        nodeptr p;
        intch,i,pos;
        p=list->next;
        printf("\n1.delete at begining\n2.delete at given pos\n");
        printf("\nenter your choice:");
        scanf("%d",&ch);
```

```
        switch(ch)
        {
                case 1: list->next=p->next;
                        p=p->next;
                        break;
                case 2:         printf("\nEnter the position for delete:");
                        scanf("%d",&pos);
                        for(i=1;i<pos-1;i++)
                                p=p->next;
                        p->next=p->next->next;
                        break;
                default:printf("\nwrong choice");
        }
}
```

Output:

```
**************
MENU
***********

1.create
2.insert
3.delete
4.display
5.exit

enter your choice:1

Enter the at end -999

Enter the number:23-999

Enter the number:
elements are:23-->
**************
MENU
***********

1.create
```

```
2.insert
3.delete
4.display
5.exit

enter your choice:2

enter the insert number:56

1.begining
2.given position
enter your choice:1

elements are:56-->23-->
**********
MENU
**********

1.create
2.insert
3.delete
4.display
5.exit

enter your choice:2

enter the insert number:34

1.begining
2.given position
enter your choice:2

Enter the position to insert:1

elements are:56-->34-->23-->
**********
MENU
**********

1.create
```

```
2.insert
3.delete
4.display
5.exit

enter your choice:3

1.delete at begining
2.delete at given pos

enter your choice:1

elements are:34-->23-->
**********
MENU
**********

1.create
2.insert
3.delete
4.display
5.exit

enter your choice:5

--------------------------------
Process exited after 43.65 seconds with return value 0
Press any key to continue . . .
```

Exercise :19

Aim: Implement a binary search tree traversal and print height of the binary search tree.

Program :

```
#include<stdio.h>
struct node
{
        int info;
        struct node *left,*right;
```

```
};
typedefstruct node *nodeptr;
nodeptrgetnode();
nodeptr create(nodeptr);
void insert(nodeptr, nodeptr);
void inorder(nodeptr);
void preorder(nodeptr);
void postorder(nodeptr);
intbtheight(nodeptr);
main()
{
        nodeptr tree;
        intch;

        tree=NULL;
        tree=create(tree);
        while(1)
        {
                printf("\n**********\n\n\tMENU\n");
                printf("\n**********\n\n1.in order\n2.pre order\n");
                printf("\n3.post order\n4. Height of BST\n5.exit\nenter your choice:");
                scanf("%d",&ch);
                switch(ch)
                {
                        case 1:         printf("\nelements in in oreder is:\n");
                                inorder(tree);
                                break;
                        case 2:         printf("\nelements in pre order is:\n");
                                preorder(tree);
                                break;
                        case 3:         printf("\nelements in post order is:\n");
                                postorder(tree);
                                break;
                        case 4: printf("Height of the Binary Search
Tree:%d",btheight(tree));
                                break;
                        case 5: exit(0);
                }
        }
}
```

```
nodeptrgetnode()
{
        nodeptr p;
        p=(nodeptr)malloc(sizeof(struct node));
        p->info=0;
        p->left=p->right=NULL;
        return p;
}
nodeptrcreate(nodeptr p)
{
        nodeptr temp;
        int k;
        temp=getnode();
        printf("\nEnter at end -999\n");
        printf("\nEnter the number:");
        scanf("%d",&k);
        temp->info=k;
        while(temp->info!=-999)
        {
                if(p==NULL)
                        p=temp;
                else
                {
                        insert(p,temp);
                        temp=getnode();
                        printf("enter the no:");
                        scanf("%d",&k);
                        temp->info=k;
                }
        }
        return p;
}
void insert(nodeptr p, nodeptr temp)
{
        if((temp->info<p->info)&&(p->left==NULL))
                p->left=temp;
        else if((temp->info<p->info)&&(p->left!=NULL))
                insert(p->left,temp);
        else if((temp->info>p->info)&&(p->right==NULL))
                p->right=temp;
```

```
        else if((temp->info>p->info)&&(p->right!=NULL))
                insert(p->right,temp);
}

void inorder(nodeptr p)
{
        if(p!=NULL)
        {
                inorder(p->left);
                printf("%d-->",p->info);
                inorder(p->right);
        }
}
void preorder(nodeptr p)
{
        if(p!=NULL)
        {
                printf("%d-->",p->info);
                preorder(p->left);
                preorder(p->right);
        }
}
void postorder(nodeptr p)
{
        if(p!=NULL)
        {
                postorder(p->left);
                postorder(p->right);
                printf("%d-->",p->info);
        }
}
intbtheight(nodeptr p)
{
        intlh,rh;
        if(p==NULL)
                return 0;
        else
        {
                lh=btheight(p->left);
                rh=btheight(p->right);
```

```
            if(lh<rh)
                return rh+1;
            else
                return lh+1;
        }
}
```

Out put

```
Enter at end -999

Enter the number:56
enter the no:12
enter the no:34
enter the no:67
enter the no:89-999
enter the no:
**********

      MENU

**********

1.in order
2.pre order

3.post order
4. Height of BST
5.exit
enter your choice:4
Height of the Binary Search Tree:3
**********

      MENU

**********

1.in order
2.pre order

3.post order
```

4. Height of BST
5.exit
enter your choice:1

elements in in oreder is:
12-->34-->56-->67-->89-->
****

MENU

****

1.in order
2.pre order

3.post order
4. Height of BST
5.exit
enter your choice:2

elements in pre order is:
56-->12-->34-->67-->89-->
****

MENU

****

1.in order
2.pre order

3.post order
4. Height of BST
5.exit
enter your choice:3

elements in post order is:
34-->12-->89-->67-->56-->
****

MENU

****

1.in order
2.pre order

3.post order
4. Height of BST
5.exit
enter your choice:5