Data Structure (01CE1301)

Lab Manual (2023-24)

Name: KISHANKUMAR VYAS
ER no.: 92200103183
Class: CE TC3 (A)

INDEX

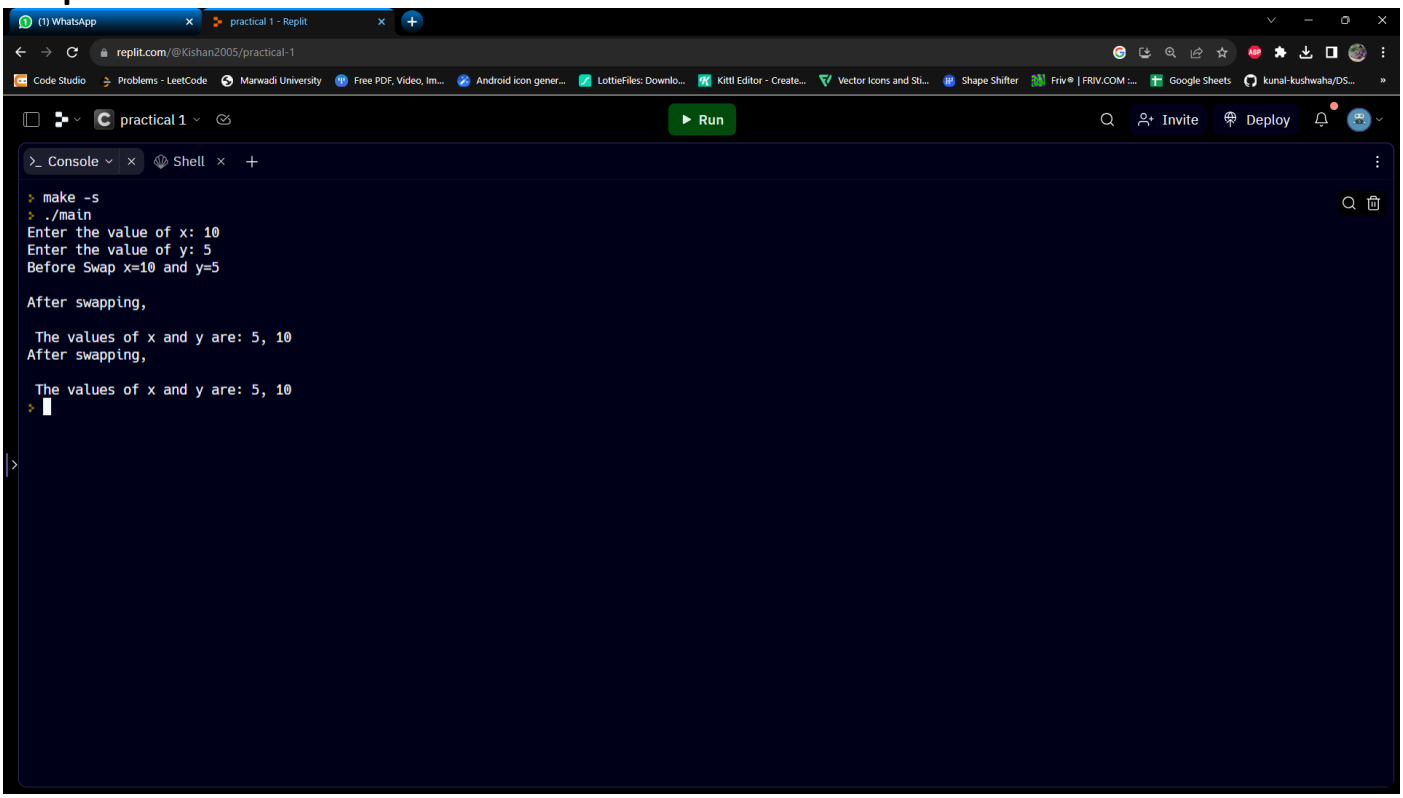
Practical 1: Introduction to pointers. Implement Call by value and Call by Reference.

Code Snippet:

```
#include <stdio.h>
void swap1(int, int);
void swap2(int*, int*);

 int main()
 // function to swap variables int x, y;
 {
  int x,y;
  printf("Enter the value of x: ");
  scanf("%d", &x);
  printf("Enter the value of y: ");
  scanf("%d", &y);
  printf("Before Swap x=%d and y=%d\n\n", x,y);
  swap1(x, y);
  swap2(&x,&y);
  return 0;
 }
 void swap1(int a, int b)
 {
  int temp;
  temp = a;
  a = b;
  b = temp;
  printf("After swapping,\n\n The values of x and y are: %d, %d\n",a,b);
 }
 void swap2(int *a, int *b)
 {
  int temp;
  temp = *a;
  *a = *b;
  *b = temp;
  printf("After swapping,\n\n The values of x and y are: %d, %d\n",*a,*b);
 }
```

Output:


(1) WhatsApp
practical 1 - Replit
replit.com/@Kishan2005/practical-1
practical 1
Run
Invite
Deploy
Console
Shell
make -s
./main
Enter the value of x: 10
Enter the value of y: 5
Before Swap x=10 and y=5
After swapping,
The values of x and y are: 5, 10
After swapping,
The values of x and y are: 5, 10

Practical 2: Write a program to implement STACK using array that performs following operations: (a) PUSH (b) POP (c) PEEP (d) CHANGE (e) DISPLAY.

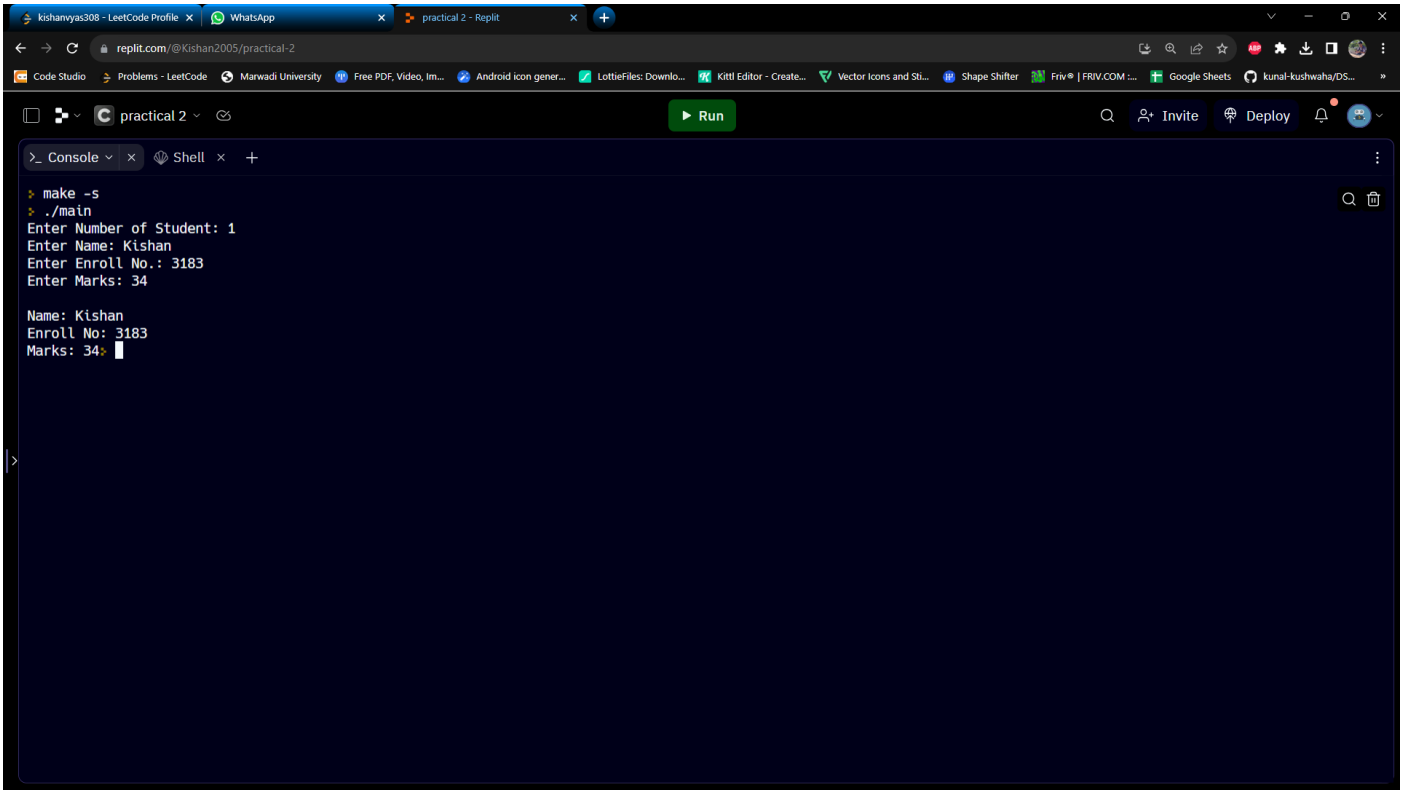
Code Snippet:

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

struct student
{
    int enroll;
    int marks;
    char name[20];
};
void main()
{
 int n,i;
    struct student *p;
    printf("Enter Number of Student: ");
    scanf("%d",&n);
    p=(struct student*)calloc(n,sizeof(struct student));
    if(p==NULL)
    {
        printf("Memory is not available");
    }
    else
    {
        for(i=0;i<n;i++)
        {
            printf("Enter Name: ");
            scanf("%s",&p[i].name);
            printf("Enter Enroll No.: ");
            scanf("%d",&p[i].enroll);
            printf("Enter Marks: ");
            scanf("%d",&p[i].marks);
        }
        for(i=0;i<n;i++)
        {
            printf("\nName: %s",p[i].name);
            printf("\nEnroll No: %d",p[i].enroll);
            printf("\nMarks: %d",p[i].marks);
        }
```

```
            free(p);
        return 0;
       }
}
```

Output:



Practical 3: Introduction to Dynamic Memory Allocation and use of DMA functions malloc(), calloc(), free(), etc.

Code Snippet:

```
#include <stdio.h>
#include <stdlib.h>
#define MAX 5

int S[MAX], top=-1;
int isFull()
{
 if(top==MAX-1)
   return 1;
 else
   return 0;
}

void push(int x)
{
 if(isFull())
   printf("Stack is overflow");
 else{
   top++;
   S[top]=x;
   printf("Value pushed successfully");
 }
}

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

void display(){
 int i;
 if(isEmpty())
   printf("Stack is Empty");
 else{
   printf("stack is: ");
   for(i=top;i>=0;i--)
     printf("%d",S[i]);
 }
```

```
}

void peep(){
 if(isEmpty())
  printf("Stack is Empty");
 else{
  printf("Top Most Element is %d",S[top]);

 }
}

void pop()
{
   if(isEmpty())
   {
      printf("underflow\n");
   }
   else
   {
      printf("element deleted");
      top--;
   }
}


void main()
{
int ch,v;

do
 {
   printf("\nstack operation");
   printf("1.push\n2.pop\n3.peep\n4.display\n5.isEmpty\n6.isFull\n7.Exit");
   printf("\nEnter Your choice : ");
   scanf("%d",&ch);
   switch(ch)
    {
    case 1: printf("Enter a Value : ");
    scanf("%d",&v);
    push(v);
    break;
    case 2: pop(); break;
    case 3: peep(); break;
    case 4: display(); break;
    case 5: if(isEmpty())
    printf("Yes,stack is empty");
    else
    printf("No,stack is not empty");
```

```
    break;
    case 6: if(isFull())
    printf("Yes,stack is Full");
    else
    printf("No,stack is not Full");
    break;
    case 7:exit(0);
      break;
    default: printf("\nInvalid Choice");
   }
 }while(ch!=7);

}
```

Output: