WINTER SEMESTER 2016

CSE2003: DATA STRUCTURES AND ALGORITHMS (EMBEDDED LAB) SLOT:

L51+L52

FACULTY: THENDRAL.P

ASSIGNMENT-1

Name: VOLETI RAVI

21. Create a structure for books (book title, author, pages, year) and stack the books one over the other such that the resulting stack has the books arranged in the order of year of publication. (Don't do sorting of the stack)

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

struct books
{
        char title[40];
        char author[25];
        int pages;
        int year;
};

struct books A[50];
int top = -1;

void push()
```

```
{
        if(top==49)
        {
                printf("The stack is full\n");
                return;
        }
        else if(top==-1)
        {
        top = top+1;
        printf("Enter the title of the book");
        gets(A[top].title);
        printf("Enter the author of the book");
        gets(A[top].author);
        printf("Enter the pages of the book");
        scanf("%d",&A[top].pages);
        printf("Enter the year of the book");
        scanf("%d",&A[top].year);
        }
        else
        {


        }
}

int pop()
{
        if(top==-1)
        {
                printf("The stack is empty\n");
                return 0;
        }
```

```
        return A[top--];
}

 int main()
 {

 }
```

OUTPUT:

22. Create a structure Job (job title, file type, size, author). Write a C program to implement the scheduling of jobs to a printer on the basis of first come first serve. Provide options to add a job, cancel a job and display the status of jobs

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

struct Job
{
        char title[25];
        char type[20];
        float size;
        char author[30];
        struct Job* next;
};

struct Job* head = NULL;
```

```
struct Job* rear = NULL;


void add()
{
	struct Job* temp;
	temp = (struct Job*)malloc(sizeof(struct Job));
	printf("Enter the title\n");
	fflush(stdin);
	gets(temp->title);
	printf("Enter the type of Job\n");
	fflush(stdin);
	gets(temp->type);
	printf("Enter the size of the book\n");
	scanf("%f",&temp->size);
	printf("Enter the author of book\n");
	fflush(stdin);
	gets(temp->author);
	temp->next = NULL;
	if(head==NULL&&rear==NULL)
	{
		head = rear = temp;
		return ;
	}
	rear->next = temp;
	rear = temp;
}

void cancel()
{
	printf("Enter the title of the job u need to cancel\n");
	char t[25];
```

```
	fflush(stdin);
	gets(t);
	struct Job* temp = head;
	if(strcmp(t,head->title)==0)
	{
		head = head->next;
		free(temp);
		return ;
	}
	struct Job* temp1 = head->next;
	while(temp1!=rear)
	{
		if(strcmp(t,temp1->title)==0)
		{
				break;
		}
		temp = temp->next;
		temp1 = temp1->next;
	}
	temp->next = temp1->next;
	free(temp1);
}

void print()
{
	struct Job* temp = head;
	while(temp!=NULL)
	{
		printf("Title %s",temp->title);
		printf("\nType %s",temp->type);
		printf("\nSize %f",temp->size);
```

```
            printf("\nAuthor %s",temp->author);
            temp = temp->next;
        }
}

int main()
{
        add();
        add();
        print();
        cancel();
        print();
}
```

OUTPUT: