PPS Assignment-3

Submitted by: Parijat Kumar

Year: 1st

Branch: Computer engineering with Specialization in Data science

Roll No.: 20001016037

Submitted to: Alka Ma'am

1. Create a structure to specify data of customers in a bank. The data to be stored is: Account number, Name, Balance in account. Assume maximum of 200 customers in the bank.
(a) Write a function to print the Account number and name of each customer with balance below Rs. 100.
(b) If a customer request for withdrawal or deposit, it is given in the form: Acct. no, amount, code (1 for deposit, 0 for withdrawal)
Write a program to give a message, "The balance is insufficient for the specified withdrawal".

Program:

Q1.c > main()

```c
#include<stdio.h>
int action(int, int, int);
int below100();
struct acc_holder
{
    long int acc_num;
    char name[30];
    int bal;
}
sbi[200] = {
1, "Aditya", 17042001,
2, "Himanshu", 12092001,
3, "Tushar", 18072002,
4, "Mehak", 13072002,
5, "Parijat", 7};
int main()
{
    printf("Name: Parijat Kumar\nRoll No.: 20001016037\n***************\n");
    int accnum, amount, code;
    printf("\nEnter your account number : ");
    scanf("%d", &accnum);
    printf("Enter 1 for deposit and 0 for withdrawal : ");
    scanf("%d", &code);
    if (code==1)
    {
        printf("\nEnter amount to be deposit : ");
        scanf("%d", &amount);
    }
    else if(code==0)
    {
        printf("\nEnter amount to withdraw : ");
        scanf("%d", &amount);
    }
    else
    {
        printf("Enter a valid number.");
    }
    action(accnum, amount, code);
    printf("All members with account balance less than 100 are following : \n");
    below100();
    return 0;
}
```

```c
int below100()
{
    int i;
    for (i = 0; i < 200; i++)
    {
        if (sbi[i].bal < 100 && sbi[i].bal > 0)
        {
            printf("\nName : %s\n", sbi[i].name);
            printf("\nAccount Number : %d\n", sbi[i].acc_num);
        }
    }
    return 0;
}
int action(int accnum, int amount, int code)
{
    int i;
    for (i = 0; i < 200; i++)
        if (sbi[i].acc_num == accnum)
            break;
    if (!code)
    {
        if (sbi[i].bal - amount < 100)
        {
            printf("\nThe balance is insufficient for the specified withdrawal\n");
        }
        else
        {
            sbi[i].bal -= amount;
            printf("\nYour new account balance is : %d\n", sbi[i].bal);
        }
    }
    else
    {
        sbi[i].bal += amount;
        printf("\nYour new account balance is : %d\n", sbi[i].bal);
    }
    return 0;
}
```

Output:

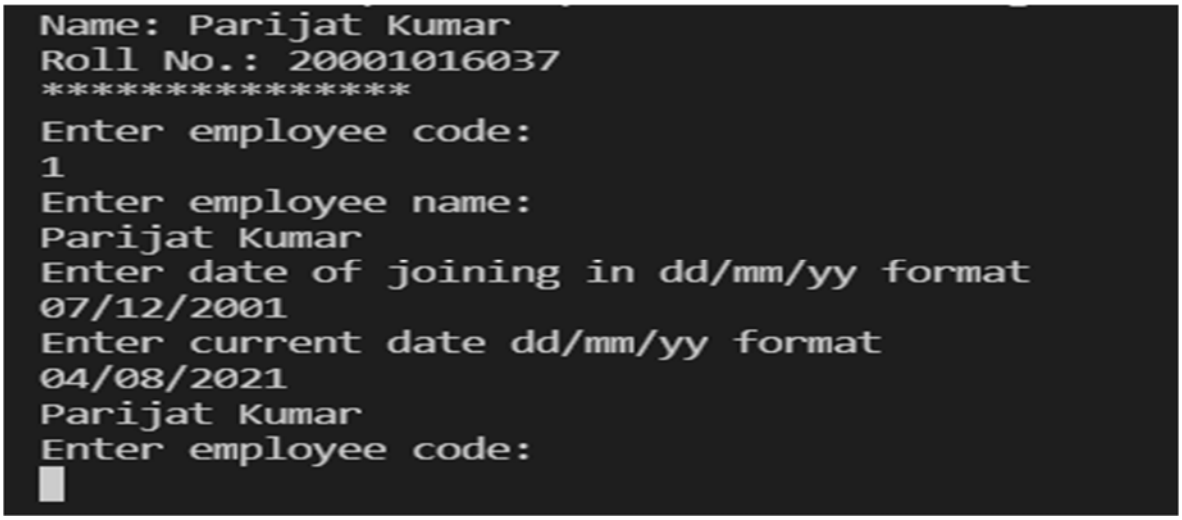
```
Name: Parijat Kumar
Roll No.: 20001016037
***************

Enter your account number : 5
Enter 1 for deposit and 0 for withdrawal : 0

Enter amount to withdraw : 100

The balance is insufficient for the specified withdrawal
All members with account balance less than 100 are following :

Name : Parijat

Account Number : 5
```

2. There is a structure called **employee** that holds information like employee code, name, date of joining. Write a program to create an array of the structure and enter some data into it. Then ask the user to enter current date. Display the names of those employees whose tenure is 3 or more than 3 years according to the given current date.

Program:

```c
#include<stdio.h>
#include<string.h>
int main()
{
    struct employee
    {
        char name[40];
        int code,doj,moj,yoj;
    };
    struct employee e[3];
    int i,d,m,y,yr;
    printf("Name: Parijat Kumar\nRoll No.: 20001016037\n***************\n");
      for(i=0;i<3;i++)
    {
        printf("Enter employee code: \n");
        scanf("%d",&e[i].code);
        fflush(stdin);
        printf("Enter employee name: \n");
        gets(e[i].name);
        printf("Enter date of joining in dd/mm/yy format \n");
        scanf("%d/%d/%d",&e[i].doj,&e[i].moj,&e[i].yoj);
        printf("Enter current date dd/mm/yy format\n");
        scanf("%d/%d/%d",&d,&m,&y);
        yr=y-e[i].yoj;
        if(yr>3)
        {
            printf("%s \n",e[i].name);
            continue;
        }
        if(yr==3)
        {
            if(e[i].moj>m)
            {
                printf("%s \n",e[i].name);
                continue;
            }
        }
        if(e[i].moj==m)
        {
            if(e[i].doj>=d)
            {
                printf("%s \n",e[i].name);
            }
        }
    }
    return 0;
}
```

Output:


Name: Parijat Kumar
Roll No.: 20001016037
****
Enter employee code:
1
Enter employee name:
Parijat Kumar
Enter date of joining in dd/mm/yy format
07/12/2001
Enter current date dd/mm/yy format
04/08/2021
Parijat Kumar
Enter employee code:

3. Write a menu driven program that depicts the working of a library. The menu options should be:

1. Add book information
2. Display book information
3. List all books of given author
4. List the title of specified book
5. List the count of books in the library
6. List the books in the order of accession number
7. Exit

Create a structure called **library** to hold accession number, title of the book, author name, price of the book, and flag indicating whether book is issued or not.

Program:

Q3.c > main()

```c
#include<stdio.h>
int main()
{
    struct lib
    {
        int accession,flag;
        float price;
        char name[20],authname[20];
    };
    struct lib l[20];
    int ch,i=0,acc,j=0,x;
    char author[20];
    printf("Name: Parijat Kumar\nRoll No.: 20001016037\n***************\n");
    while(x=1)
    {
    printf("What do u want to do? \n");
    printf("1.Add book information \n");
    printf("2.Display book information \n");
    printf("3.List all books of given author \n");
    printf("4.List the title of specified book \n");
    printf("5.List the count of books in the library \n");
    printf("6.List the books in the order of accession no. \n");
    printf("7.Exit\n");
    scanf("%d",&ch);
    switch(ch)
    {
        case 1:
        {
            printf("Enter the name of the book: \n");
        fflush(stdin);
            gets(l[j].name);
             printf("Enter the author name: \n");
         fflush(stdin);
            gets(l[j].authname);
            printf("Enter the price of the book: \n");
            scanf("%f",&l[j].price);
            printf("press 0 if book is issued and 1 if it is available: \n");
            scanf("%d",&l[j].flag);
            printf("Record added successfully\n");
            l[j].accession=j;
            j++;
            break;
        }
        case 2:
        {
           for(i=0;i<j;i++)
               puts(l[i].name);
            printf(" ");
            puts(l[i].authname);
            printf("%f ",l[i].price);
            if(l[i].flag==0)
                printf("Book is available\n");
            else
                printf("Book is not available\n");
            break;
        }
```

```
            case 3:
            {
                printf("Enter the name of author\n");
                gets(author);
                for(i=0;i<j;i++)
                {
                    if(l[i].authname==author)
                    {
                        puts(l[i].name);
                        printf("\n");
                    }
                }
                break;
            }
            case 4:
            {
                printf("Enter the accession no. of the book\n");
                scanf("%d",&acc);
                for(i=0;i<j;i++)
                    if(l[i].accession==acc)
                    {
                        puts(l[i].name);
                        printf("\n");
                    }
                break;
            }
            case 5:
            {
                printf("%d",j);
                break;
            }
            case 6:
            {
                for(i=0;i<j;i++)
                {
                    puts(l[i].name);
                    printf("\n");
                }
                break;
            }
            case 7:
            {
                x=2;
                break;
            }
        }
        }
        return 0;
    }
```

Output:

```
Name: Parijat Kumar
Roll No.: 20001016037
***************
What do u want to do?
1.Add book information
2.Display book information
3.List all books of given author
4.List the title of specified book
5.List the count of books in the library
6.List the books in the order of accession no.
7.Exit
1
Enter the name of the book:
Let Us C
Enter the author name:
Yashwant Kanetkar
Enter the price of the book:
700
press 0 if book is issued and 1 if it is available:
1
Record added successfully
```

4. Write a program that compares two given dates. To store date use structure say **date** that contains three members namely date, month and year. If the dates are equal then display message as "Equal" otherwise "Unequal".

Program:

```c
#include<stdio.h>
struct date
{
    int day;
    int month;
    int year;
};
int main()
{
    struct date d1,d2;
    printf("Name: Parijat Kumar\nRoll No.: 20001016037\n**************\n");
    printf("Enter first date in the dd/mm/yyyy format: \n");
    scanf("%d/%d/%d",&d1.day,&d1.month,&d1.year);
    printf("\nEnter second date in the dd/mm/yyyy format: \n");
    scanf("%d/%d/%d",&d2.day,&d2.month,&d2.year);
    if((d1.day==d2.day)&&(d1.month==d2.month)&&(d1.year==d2.year))
    {
        printf("\nEQUAL");
    }
    else
    {
        printf("\nUNEQUAL");
    }
return 0;
}
```

Output:

```
Name: Parijat Kumar
Roll No.: 20001016037
**************
Enter first date in the dd/mm/yyyy format:
07/12/2001

Enter second date in the dd/mm/yyyy format:
07/12/2001

EQUAL
```

5. Write a program using 15 built in string functions.

Program:

```c
#include<stdio.h>
#include<string.h>
int main()
{
    char str1[70]="Parijat", str2[70]="Kumar", name[70], *s1 = "Duplicated",*s2;
    printf("Name: Parijat Kumar\nRoll No.: 20001016037\n**************\n");
    puts("Enter your name:");
    gets(name);
    puts(name);
    printf("The length of first string is %d\n",strlen(str1));
    printf("The first string in lowercase is %s\n",strlwr(str1));
    printf("The first string in uppercase is %s\n",strupr(str1));
    if (strcmp(str1, str2) ==0)
    {
        printf("String 1 and String 2 are equal\n");
    }
     else
    {
        printf("String 1 and String 2 are different\n");
    }
    strcat(str1,str2);
    printf("Output String after concatenation: %s\n", str1);
    strcpy(str1,str2);
    printf("String 1 is now : %s\n", str1);
    printf ("The character 'K' is found in string 1 at %s\n", strchr(str1, 'K'));
    printf("The occurence of string 'Kum' is found in string at %s\n",strstr(name, "Kum" ));
    printf("String entered is: %s\n",name);
    printf("The reverse of that string is: %s\n",strrev(name));
    s2 = strdup(s1);
    printf("Duplicated string is : %s\n", s2);
    printf("The name string now is %s\n",strset(name,'#'));
    return 0;
}
```

Output:

```
Name: Parijat Kumar
Roll No.: 20001016037
**********
Enter your name:
Parijat Kumar
Parijat Kumar
The length of first string is 7
The first string in lowercase is parijat
The first string in uppercase is PARIJAT
String 1 and String 2 are different
Output String after concatenation: PARIJATKumar
String 1 is now : Kumar
The character 'K' is found in string 1 at Kumar
The occurence of string 'Kum' is found in string at Kumar
String entered is: Parijat Kumar
The reverse of that string is: ramuK tajiraP
Duplicated string is : Duplicated
The name string now is ###############
```

6. Write a program that replaces two or more consecutive blanks in a string by a single blank. For example, if the input is

Grim return to the planet of apes!!

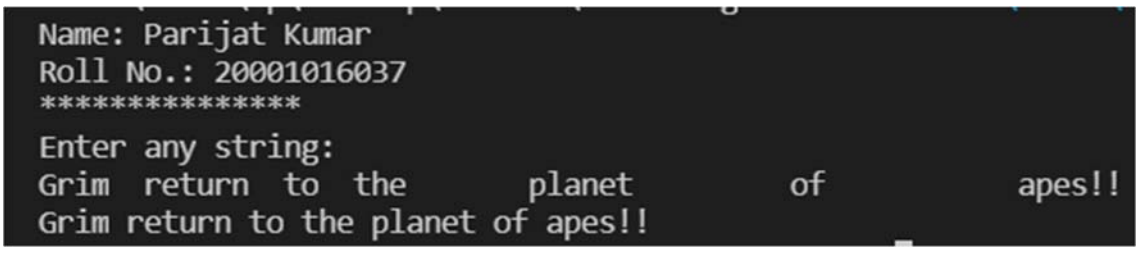
the output should be

Grim return to the planet of apes!!

Program:

```c
#include <stdio.h>
#include <string.h>
int main()
{
    char s[700];
    int i=0,count=0;;
    printf("Name: Parijat Kumar\nRoll No.: 20001016037\n**************\n");
    printf("Enter any string: \n");
    gets(s);
    while(s[i]!='\0')
    {
        if(s[i]==' ')
        {
            count++;
            i++;
            continue;
        }
        if(count>1)
        {
            printf(" %c",s[i]);
            count=0;
        }
        else
        {
            printf("%c",s[i]);
        }
        i++;
    }
    return 0;
}
```

Output:



```
Name: Parijat Kumar
Roll No.: 20001016037
**************
Enter any string:
Grim  return  to  the      planet       of          apes!!
Grim return to the planet of apes!!
```

7. Write a program to carry out the following:
(a) Read a text file 'INPUT.TXT'
(b) Print each word in reverse order
Example,
Input: INDIA IS MY COUNTRY
Output: AIDNI SI YM YRTNUOC
Assume that each word length is maximum of 10 characters and each word is separated by newline/blank characters.

Program:

```c
#include<stdio.h>
#include<string.h>
int main()
{
    printf("Name: Parijat Kumar\nRoll No.: 20001016037\n**************\n");
    FILE *fs;
    char s[80];
    void rev();
    fs=fopen("INPUT.TXT","r");
    if(fs==NULL)
    {
        printf("File cannot be opened.");
    }
    while(fgets(s,79,fs)!=NULL)
        rev(s);
        fclose(fs);
    return 0;
}
void rev(char s1[80])
{
    char s2[80];
    int i=0,j=0;
    while(s1[i]!='\0')
    {
        s2[j]=s1[i];
        if(s1[i]==' ' || s1[i]=='\0')
        {
            s2[j]='\0';
            strrev(s2);
            printf("%s ",s2);
            j=-1;
        }
        i++;
        j++;
    }
    s2[j]='\0';
    printf("%s",strrev(s2));
}
```

Output:

```
INPUT.txt
1    INDIA IS MY COUNTRY
```

```
Name: Parijat Kumar
Roll No.: 20001016037
**************
AIDNI SI YM YRTNUOC
```

8. Given a text file, write a program to create another text file deleting the words “a”, “the”, “an” and replacing each one of them with a blank space.

Program:

Q8.c > replace(char *, char *)

```c
#include<stdio.h>
#include<string.h>
void replace();
int main()
{
    printf("Name: Parijat Kumar\nRoll No.: 20001016037\n***************\n");
    FILE *fp,*ft;
    char str[80],target[80];
    fp=fopen("FILE.TXT","r");
    if(fp==NULL)
    {
        puts("Source file cannot be opened.");
    }
    ft=fopen("NEW.TXT","w");
    if(ft==NULL)
    {
        puts("Target file cannot be opened.");
    }
    while(fgets(str,79,fp)!=NULL)
    {
        replace(str,&target);
        fputs(target,ft);
    }
    fclose(fp);
    fclose(ft);
    printf("\nTask completed!\n");
    return 0;
}
void replace(char *s, char *s1)
{
        int i=0,j=0,k=0;
        char temp[100],temp2[100],main[100],*t=temp,*m=main;
        while(*s!='\0')
        {
            *t=*s;
            t++;
            s++;
        }
        *t='\0';
        while(temp[i]!='\0')
        {
            temp2[j]=temp[i];
        if(temp[i]==' ')
        {
            temp2[j]='\0';
            if(strcmpi(temp2,"the")==0)
            {
                strcpy(temp2," ");
            }
            else if(strcmpi(temp2,"an")==0)
            {
                strcpy(temp2," ");
            }
```

```
54              else if(strcmpi(temp2,"a")==0)
55              {
56                  strcpy(temp2," ");
57              }
58              j=0;
59              while(temp2[j]!='\0')
60              {
61                  main[k]=temp2[j];
62                  k++;
63                  j++;
64          }
65              main[k]=' ';
66              k++;
67              j=-1;
68          }
69          i++;
70          j++;
71          }
72          temp2[j]='\0';
73          if(strcmpi(temp2,"the")==0)
74          {
75              strcpy(temp2," ");
76          }
77          else if(strcmpi(temp2,"an")==0)
78          {
79              strcpy(temp2," ");
80          }
81          else if(strcmpi(temp2,"a")==0)
82          {
83              strcpy(temp2," ");
84          }
85          else
86          {
87              j=0;
88              while(temp2[j]!='\0')
89              {
90                  main[k]=temp2[j];
91                  k++;
92                  j++;
93              }
94              main[k]='\0';
95          }
96          while(*m!='\0')
97          {
98              *s1=*m;
99              s1++;
100             m++;
101         }
102         *s1='\0';
103 }
```

Output:


FILE.txt
1 The quick brown fox jumps over a lazy dog


Name: Parijat Kumar
Roll No.: 20001016037
****
Task completed!


NEW.txt
1 quick brown fox jumps over lazy dog

9. A stack is a data structure in which addition of new element or deletion of existing element always takes place at the same end. This end is often known as 'top' of stack. This situation can be compared to a stack of plates in a cafeteria where every new plate taken off the stack is also from the 'top' of the stack. There are several applications where stack can be put to use. For example, recursion, keeping track of function calls, evaluation of expressions, etc. Write a program to implement a stack using a linked list.

Program:

Q9.c > main()

```c
#include<stdio.h>
#include <stdlib.h>
void push();
void pop();
void display();
struct node
{
int val;
struct node *next;
};
struct node *head;
int main ()
{
    int choice=0;
    printf("Name: Parijat Kumar\nRoll No.: 20001016037\n**************\n");
    printf("\n*********Stack operations using linked list*********\n");
    printf("\n-----------------------------------------------\n");
    while(choice != 4)
    {
        printf("\n\nChose one from the below options...\n");
        printf("\n1.Push\n2.Pop\n3.Show\n4.Exit");
        printf("\n Enter your choice \n");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1:
            {
                push();
                break;
            }
            case 2:
            {
                pop();
                break;
            }
            case 3:
            {
                display();
                break;
            }
            case 4:
            {
                printf("Exiting....");
                break;
            }
            default:
            {
                printf("Please Enter valid choice.\n");
            }
        };
    }
    return 0;
}
```

```c
void push ()
{
    int val;
    struct node *ptr = (struct node*)malloc(sizeof(struct node));
    if(ptr == NULL)
    {
        printf("not able to push the element");
    }
    else
    {
        printf("Enter the value\n");
        scanf("%d",&val);
        if(head==NULL)
        {
            ptr->val = val;
            ptr -> next = NULL;
            head=ptr;
        }
        else
        {
            ptr->val = val;
            ptr->next = head;
            head=ptr;

        }
        printf("Item pushed");

    }
}
void pop()
{
    int item;
    struct node *ptr;
    if (head == NULL)
    {
        printf("Underflow\n");
    }
    else
    {
        item = head->val;
        ptr = head;
        head = head->next;
        free(ptr);
        printf("Item popped\n");
    }
}
```

```c
void display()
{
    int i;
    struct node *ptr;
    ptr=head;
    if(ptr == NULL)
    {
        printf("Stack is empty\n");
    }
    else
    {
        printf("Printing Stack elements \n");
        while(ptr!=NULL)
        {
            printf("%d\n",ptr->val);
            ptr = ptr->next;
        }
    }
}
```

Output:

```
Name: Parijat Kumar
Roll No.: 20001016037
***************

*********Stack operations using linked list*********

-----------------------------------------------


Chose one from the below options...

1.Push
2.Pop
3.Show
4.Exit
 Enter your choice
1
Enter the value
90
Item pushed

Chose one from the below options...

1.Push
2.Pop
3.Show
4.Exit
 Enter your choice
```

10. Unlike a stack, in a queue the addition of new element takes place at the end (called 'rear' of queue) whereas deletion takes place at the other end (called 'front' of queue). Write a program to implement a queue using a linked list.

Program:

Q10.c > main()

```c
#include<stdio.h>
#include<stdlib.h>
struct node
{
    int data;
    struct node *next;
};
struct node *front;
struct node *rear;
void insert();
void delete();
void display();
int main ()
{
    int choice;
    printf("Name: Parijat Kumar\nRoll No.: 20001016037\n**************\n");
    while(choice != 4)
    {
        printf("\n************************Main Menu****************************\n");
        printf("\n=================================================================\n");
        printf("\n1.insert an element\n2.Delete an element\n3.Display the queue\n4.Exit\n");
        printf("\nEnter your choice ?\n");
        scanf("%d",& choice);
        switch(choice)
        {
            case 1:
            insert();
            break;
            case 2:
            delete();
            break;
            case 3:
            display();
            break;
            case 4:
            exit(0);
            break;
            default:
            printf("\nPlease enter a valid number.\n");
        }
    }
    return 0;
}
void insert()
{
    struct node *ptr;
    int item;

    ptr = (struct node *) malloc (sizeof(struct node));
    if(ptr == NULL)
    {
        printf("\nOVERFLOW\n");
        return;
    }
```

```
55      else
56      {
57          printf("\nEnter value?\n");
58          scanf("%d",&item);
59          ptr -> data = item;
60          if(front == NULL)
61          {
62              front = ptr;
63              rear = ptr;
64              front -> next = NULL;
65              rear -> next = NULL;
66          }
67          else
68          {
69              rear -> next = ptr;
70              rear = ptr;
71              rear->next = NULL;
72          }
73      }
74  }
75  void delete ()
76  {
77      struct node *ptr;
78      if(front == NULL)
79      {
80          printf("\nUNDERFLOW\n");
81          return;
82      }
83      else
84      {
85          ptr = front;
86          front = front -> next;
87          free(ptr);
88      }
89  }
90  void display()
91  {
92      struct node *ptr;
93      ptr = front;
94      if(front == NULL)
95      {
96          printf("\nEmpty queue\n");
97      }
98      else
99      {   printf("\nprinting values .....\n");
100         while(ptr != NULL)
101         {
102             printf("\n%d\n",ptr -> data);
103             ptr = ptr -> next;
104         }
105     }
106 }
```

Output:

```
Name: Parijat Kumar
Roll No.: 20001016037
***************

*************************Main Menu*****************************

=================================================================

1.insert an element
2.Delete an element
3.Display the queue
4.Exit

Enter your choice ?
1

Enter value?
90

*************************Main Menu*****************************

=================================================================
```