**Q 1. Write algorithm/ pseudocode/ program in programmming language you know (5 marks each)**

1. **To add first 10 terms of series using for 1 / 1! + 2/2! + 3/3! ...**

**Ans:** #include<stdio.h>

int main()

{

float answer=0,fact=1;

for(int i=1;i<=10;i++)

{

fact\*=i;

answer+=i/fact;

}

printf("%f",answer);

}

**b) To print first n (n is the value entered by the user) terms in Fibonacci series 1,1,2,3,5,8,13..... For example, if user enters n = 5, first 5 terms are displayed.**

**Ans:** int main()

{

int n;

printf("Enter n: ");

scanf("%d",&n);

if(n==1)

printf("1");

else if(n==2)

printf("1 1 ");

else

{

int first=1;

int second=1;

printf("%d %d ",first, second);

for(int i=2;i<=n;i++)

{

int temp=first+second;

printf("%d ",temp);

first=second;

second=temp;

}

}

}

**c) Display following menu and execute corresponding operation till user selects option 4 i.e. Exit -**

**1. Input a matrix**

**2. Display a matrix**

**3. Multiply a matrix with given number**

**4. Exit**

**Implement functions of carryout first three operations**.

**Ans**: #include<stdio.h>

int m,n;

int k;

int a[100][100], b[100][100];

void ReadMatrix(){

printf("\nEnter m and n: ");

scanf("%d %d",&m,&n);

for(int i=0;i<m;i++){

for(int j=0;j<n;j++){

scanf("%d",&a[i][j]);

}

}

}

void Display(){

int c;

printf("\n1. Display input\n 2. Display output");

scanf("%d",&c);

switch(c){

case 1: for(int i=0;i<m;i++)

{

for(int j=0;j<n;j++){

printf("%d ",a[i][j]);

}

printf("\n");

}

break;

case 2: for(int i=0;i<m;i++)

{

for(int j=0;j<n;j++){

printf("%d ",b[i][j]);

}

printf("\n");

}

break;

}

}

void Multiply(){

printf("\nEnter k: ");

scanf("%d",&k);

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

{

for(int j=0;j<n;j++)

{

b[i][j]=k\*a[i][j];

}

}

}

int main(){

int choice;

do{

printf("Enter the options\n ");

printf("1. Read Matrix\n 2. Display the matrix\n3. Multiply the matrix\n4. Exit");

scanf("%d",&choice);

switch(choice){

case 1: ReadMatrix();

break;

case 2: Display();

break;

case 3: Multiply();

break;

case 4: exit(0);

}

}while(choice!=4);

return 0;

}

**d) Accept a number and print its digits in words. No = 235 - two three five.**

**Ans:** int main()

{

int n;

printf("Enter n: ");

scanf("%d",&n);

int sum=0,r=0;

while(n>0)

{

r=n%10;

sum=sum\*10+r;

n=n/10;

}

n=sum;

while(n>0)

{

r=n%10;

switch(r)

{

case 1:

printf("one ");

break;

case 2:

printf("two ");

break;

case 3:

printf("three ");

break;

case 4:

printf("four ");

break;

case 5:

printf("five ");

break;

case 6:

printf("six ");

break;

case 7:

printf("seven ");

break;

case 8:

printf("eight ");

break;

case 9:

printf("nine ");

break;

case 0:

printf("zero ");

break;

default:

printf("invalid ");

break;

}

n=n/10;

}

}

**e) To read a sentence and word, check if word exists in sentence, if yes then display words position in sentence**

**Ans:** #include<stdio.h>

#include<string.h>

int main()

{

char str[100], word[100];

//Reading the input

printf("Enter string: ");

scanf("%[^\n]s",str);

getchar();

printf("\nEnter word: ");

scanf("%[^\n]s",word);

getchar();

int n=strlen(str);

char temp[n];

int flag=0;

int pos=1, j=0;

strcpy(temp,"");

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

{

if(str[i]==' '||str[i]=='\0')

{

temp[j++]='\0';

if(strcmp(temp,word)==0)

{

flag=1;

break;

}

strcpy(temp,"");

pos++;

}

else

{

strncat(temp,&str[i],1);

j++;

}

}

if(flag==1)

printf("Word Found at position=%d",pos);

else

printf("Word Not Found");

return 0;

}

2.a) Classes:

1. Employee
2. Project
3. Bugs
4. Ticket
5. Note

2.b)

|  |  |
| --- | --- |
| Class Name | Employee |
| Attributes | * EmployeeID * employeeName * employeeMail * Password |
| Functions | * Login() * RaiseTicket() * AddNote() * PasteCode() * Tag() * CreateEmployee() * DisplayEmployee() * EditEmployee() * DeleteEmployee() * CloseTicket() |

|  |  |
| --- | --- |
| Class Name | Project |
| Attributes | * ProjectID * ProjectName * ProjectManager * ProjectTeam * BugsCount * Bugs |
| Functions | * CreateProject() * DisplayProject() * EditProject() * DeleteProject() |

|  |  |
| --- | --- |
| Class Name | Ticket |
| Attributes | * TicketID * BugID * ProjectID * Notes |
| Functions | * AddNote() * createTicket() * EditTicket() * DisplayTicket() * CloseTicket() |

|  |  |
| --- | --- |
| Class Name | Bugs |
| Attributes | * BugID * BugName * BugLevel * BugStatus * BugEncounteredTime * BugSolvedTime |
| Functions | * CreateBug() * TrackBug() * DisplayBug() * DeleteBug() |

|  |  |
| --- | --- |
| Class Name | Note |
| Attributes | * NoteID * NoteTitle * NoteDescription |
| Functions | * HighlightKeyWords() * HighlightEmployee() * ListEmployeesToTag() * NotifyEmployee() * CreateNote() * EditNote() * DisplayNote() * DeleteNote() |

2.c)

