nidhi859 / **PPS-assignment**

No description, website, or topics provided. Edit

Manage topics

2 commits | 1 branch | 0 packages | 0 releases | 1 contributor

Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

nidhi859 Update README.md — Latest commit 5dadc7d on Nov 7

README.md — Update README.md — last month

README.md

PPS-assignment

Q1. enter 1D-array.

```c
#include<stdio.h>

int main()
{
        int arr[20],i,N;
        printf("how many integers you want to enter : ");
        scanf("%d",&N);
        printf("\nEnter Array : ");
        for(i=0;i<N;i++)
        {
                scanf("%d",&arr[i]);
        }
        printf("\n Your Array is : ");
        for(i=0;i<N;i++)
        {
                printf("%d\t",arr[i]);
        }
        return 0;
}
```

Q2. enter 2D-array.

```c
#include<stdio.h>

int main()
{
        int arr[10][10],r,c,i,j;
        printf("How many rows and column you want to enter : ");
        scanf("%d %d",&r,&c);
        printf("\nEnter The array : ");
        for(i=0;i<r;i++)
        {
                for(j=0;j<c;j++)
                {
                scanf("%d",&arr[i][j]);
            }
        }
        printf("\nYour Array is : \n");
        for(i=0;i<r;i++)
        {
                for(j=0;j<c;j++)
                {
                        printf("%d\t",arr[i][j]);
                }
                puts("\n");
        }
        return 0;
}
```

Q3. add two matrix.

```c
#include<stdio.h>
int main()
{
 int m,n,c,d,f[10][10],s[10][10],sum[10][10];
 printf("enter the number of rows and coloumns of matrix \n");
 scanf("%d%d",&m,&n);
 printf("enter the element of first matrix \n");
 for(c=0;c<m;c++)
 { for(d=0;d<n;d++)
    {
      scanf("%d",&f[c][d]);
    }
 }
   printf("enter the elements of second matrix \n");
 for(c=0;c<m;c++)
 { for(d=0;d<n;d++)
    {
      scanf("%d",&s[c][d]);
    }
 }
printf("sum of entered mateices: \n");
for(c=0;c<m;c++)
{
  for(d=0;d<n;d++)
    {
      sum[c][d]=f[c][d]+s[c][d];
      printf("%d \t",sum[c][d]);
    }
  printf("\n");
}
return 0;
}
```

Q4. print address.

```c
#include<stdio.h>

int main()
{
        puts("Sam Rajput");
        puts("House N0. 43, Moga Street");
        puts("Jalandhar, Punjab");
        puts("144602");
        return 0;
}
```

Q5. enter arry using pointers.

```c
int main()
{
int arr[10];
int *p;
int i;
p=&arr[0];
printf("enter array elements \n");
for(i=0;i<10;i++)
 { printf("enter elements %02d:\n",i+1);
 scanf("%d",p+i);
 }
printf("enetered array elements are:\n");
printf("\n address \t value \n");
for(i=0;i<10;i++)
{
 printf("%08x  \t %03d \n",(p+i),*(p+i));
}
return 0;
}
```

Q6. print area of circle.

```c
    #include<stdio.h>
#include<math.h>

int main()
{  float area,radius ;
   printf("Enter the radius of circle\n");

   scanf("%f",&radius);

   /* M_PI (pi) is a constant in math.h header file */

   area = M_PI*radius*radius;

   printf("Area of circle = %.2f\n", area);

   return 0;
}
```

Q7. binary search.

```c
     #include<stdio.h>

 int main()
 {
        int i,first,last,mid,N,arr[20],ele;
```

```
        printf("\nHow many elements you want to enter(in ascending order)  : ");
        scanf("%d",&N);
        printf("\nEnter Elements : ");
        for(i=0;i<N;i++)
        {
                scanf("%d",&arr[i]);
        }
        printf("\n Enter element you want to search : ");
        scanf("%d",&ele);
        int flag=0;
        first=0;
        last=N-1;
        while(last>=first)
        {
                mid=(first + last) / 2;
                if(arr[mid]==ele)
                {
                        printf("\n Element is found at position %d",mid+1);
                        flag=1;
                        break;
                }
                else if(arr[mid]>ele)
                {
                        last=mid-1;
                        break;
                }
                else{
                        first=mid+1;
                }
        }
        if(flag==0)
        {
                printf("\nElement Not found!.\n");
        }
        return 0;
}
```

Q9. bubble sorting.

```
#include <stdio.h>

int main()
{
  int array[100], n, c, d, swap;

  printf("Enter number of elements\n");
  scanf("%d", &n);

  printf("Enter %d integers\n", n);

  for (c = 0; c < n; c++)
    scanf("%d", &array[c]);

  for (c = 0 ; c < n - 1; c++)
  {
    for (d = 0 ; d < n - c - 1; d++)
    {
      if (array[d] > array[d+1]) /* For decreasing order use < */
      {
        swap       = array[d];
        array[d]   = array[d+1];
        array[d+1] = swap;
      }
    }
  }
```

```
  printf("Sorted list in ascending order:\n");

  for (c = 0; c < n; c++)
     printf("%d\n", array[c]);

  return 0;
}
```

Q10. calculator.

```
include<stdio.h>

int main()
{
        int num1,num2,result,choice;
        printf("Enter first Number : ");
        scanf("%d",&num1);
        printf("\nEnter second Number : ");
        scanf("%d",&num2);
        puts("Select Operation :");
        puts("1.Add \n2. Subtract\n3. Divide\n4. Multiply");
        puts("Enter Choice : ");
        scanf("%d",&choice);
        switch(choice)
        {
                case 1 : result=num1+num2;
                printf("Result : %d\n",result);
                break;
                case 2 : result=num1-num2;
                printf("Result : %d\n",result);
                break;
                case 3 : result=num1/num2;
                printf("Result : %d\n",result);
                break;
                case 4 : result=num1*num2;
                printf("Result : %d\n",result);
                break;
        }
        return 0;
}
```

Q11. program using call by value.

```
include<stdio.h>
void swap(int x,int y);
int main()
{
  int a=100,b=200;
  printf("before swaping value of a and b are %d and %d respectively \n",a,b);
 swap(a,b);
  printf("after swaping value of a and b are %d and %d respevtively \n",a,b);
return 0;
}
void swap(int x,int y)
{
  int temp;
  temp=x;
  x=y;
  y=temp;
 return ;
}
```

Q12. program using call by reference.

```c
#include<stdio.h>
void swap(int*x,int*y);
int main()
{
 int a=100,b=200;
 printf("before swaping a and b are %d and %d respectively",a,b);
 swap(&a,&b);
 printf("after swaping a and b are %d and %d respectively",a,b);
 return 0;
}
void swap(int *x,int*y)
{
  int temp;
  temp=*x;
  *x=*y;
  *y=temp;
  return;
}
```

Q13. even odd .

```c
#include<stdio.h>

int main()
{
        int num;
        printf("\n Enter an Integer : ");
        scanf("%d",&num);
        if(num%2==0)
        {
                printf("\n The integer %d is Even.",num);
        }
        else
        {
                printf("\n The integer %d is Odd.",num);
        }
        return 0;
}
```

Q14. factorial.

```c
#include<stdio.h>
int main()
{
  int c,n,fact=1;
  printf("enter a no. to calculate its factorial \n");
  scanf("%d",&n);
  for(c=1;c<=n;c++)
  {
    fact=fact*c;
  }
  printf("factorial of %d =%d\n",n,fact);
 return 0;
}
```

Q15. fibonacci series.

```c
#include<stdio.h>
int main()
{
```

```
 int n,f=0,s=1,next,c;
 printf("enter the no of terms \n");
 scanf("%d",&n);
 printf("first %d term of fibonacci series are: \n",n);
 for(c=0;c<n;c++)
 {
  if(c<=1)
   next=c;
  else
   {
     next=f+s;
     f=s;
     s=next;
   }
  printf("%d \n",next);
 }
return 0;
}
```

Q16. greatest no.

```
#include <stdio.h>

int main()
{
  int array[100], maximum, size, c, location = 1;

  printf("Enter the number of elements in array\n");
  scanf("%d", &size);

  printf("Enter %d integers\n", size);

  for (c = 0; c < size; c++)
    scanf("%d", &array[c]);

  maximum = array[0];

  for (c = 1; c < size; c++)
  {
    if (array[c] > maximum)
    {
       maximum  = array[c];
       location = c+1;
    }
  }
printf("Maximum element is present at location %d and it's value is %d.\n", location, maximum);
return 0;
}
```

Q17. hello program.

```
#include<stdio.h>

int main()
{
        puts("Hello Budding Engineers.");
        return 0;
}
```

Q18. leap year program.

```
#include<stdio.h>
int main()
{
 int year;
 printf("enter year");
 scanf("%d",&year);
 if (((year%4==0)&&(year%100!=0))||(year%400==0))
  printf("%d is a leap year",year);
 else
  printf("%d is not a leap year",year);
return 0;
}
```

Q19. linear search.

```
#include<stdio.h>
int main()
{
int arr[10],ele,i,flag=0;
printf("enter 10 integers:");
for( i=0;i<10;i++)
{
scanf("%d",&arr[i]);
}
printf("\n enter element you want to search:");
scanf("%d",&ele);
for(int i=0;i<10;i++)
{
if(arr[i]==ele)
{
printf("\n element is found :)index:%d\t position:%d\n",i,i+1);
flag=1;
break;}
}
if(flag==1)
{
printf("element is not found ");
}
return 0;
}
```

Q20. multiply matrix.

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

int main()
{
  int m, n, p, q, c, d, k, sum = 0;
  int first[10][10], second[10][10], multiply[10][10];

  printf("Enter number of rows and columns of first matrix\n");
  scanf("%d%d", &m, &n);
  printf("Enter elements of first matrix\n");

  for (c = 0; c < m; c++)
    for (d = 0; d < n; d++)
      scanf("%d", &first[c][d]);

  printf("Enter number of rows and columns of second matrix\n");
  scanf("%d%d", &p, &q);

  if (n != p)
    printf("The matrices can't be multiplied with each other.\n");
```

```
  else
  {
    printf("Enter elements of second matrix\n");

    for (c = 0; c < p; c++)
      for (d = 0; d < q; d++)
        scanf("%d", &second[c][d]);

    for (c = 0; c < m; c++) {
      for (d = 0; d < q; d++) {
        for (k = 0; k < p; k++) {
          sum = sum + first[c][k]*second[k][d];
        }
       multiply[c][d] = sum;
        sum = 0;
      }
    }

    printf("Product of the matrices:\n");

    for (c = 0; c < m; c++) {
      for (d = 0; d < q; d++)
        printf("%d\t", multiply[c][d]);

      printf("\n");
    }
  }

  return 0;
}
```

Q21. pattern 1.

```
#include<stdio.h>

int main()
{
  int  i,j,row;
  printf("Enter number of rows:");
  scanf("%d",&row);
  for(i=1;i<=row;++i)
  {
    for(j=1;j<=i;++j)
     {
       printf("* ");
     }
   printf("\n");
  }
 return 0;
}
```

Q22. pattern 2.

```
#include<stdio.h>

int main()
{
  int i,j,row;
  printf("Enter the no. of rows:");
  scanf("%d",&row);
  for(i=row;i>=1;--i)
  {
     for(j=1;j<=i;++j)
      {
```

```
        printf(" *");
       }
     printf("\n");
  }
 return 0;
}
```

Q23. pattern 3.

```
#include<stdio.h>

int main()
{
  int i,row,space,k=0;
  printf("enter no. of rows:");
  scanf("%d",&row);
  for(i=1;i<=row;++i,k=0)
  {
    for(space=1;space<=row-i;++space)
     {
       printf("   ");
     }

     while(k!=2*i-1)
     {
       printf("*  ");
       ++k;
     }
     printf("\n");
   }
return 0;
}
```

Q24. perimeter of circle.

```
      #include<stdio.h>

int main()
{
        float radius,peri;
        printf("Enter The Radius Of The  Circle : ");
        scanf("%f",&radius);
        peri=2*3.14*radius;
        printf("\nPerimeter Of The Circle : %.2f",peri);
        return 0;
}
```

Q25. pallindrom of a number.

```
#include<stdio.h>
int main()
{
  int temp,number,sum ,digit;
  printf("enter a no.");
  scanf("%d",&number);
  temp=number;
  while(temp>0)
  {
   digit=temp%10;
   temp/=10;
   sum=sum*10+digit;
  }
```

```
  if(number==sum)
    printf("entered no. is palindrom");
  else
    printf("entered no. is palindorm");
return 0;
}
```

Q26. pointer.

```
#include<stdio.h>
int main()
{
 int n;
int *p;
p=&n;
n=100;
printf("using variable n:\n");
printf("value of n:%d \n addres of n:%d\n",n,&n);
printf("using pointer value:p\n");
printf("value of n:%d\n addres of n:%d\n",*p,p);
return 0;
}
```

Q27. reverse of number.

```
    #include<stdio.h>

int main()
{
        int num,reverse=0,digit;
        printf("Enter an Integer(min. 2 digits) : ");
        scanf("%d",&num);
        int temp=num;
        while(temp>0)
        {
                digit=temp%10;
                reverse=(reverse*10)+digit;
                temp/=10;
        }
        printf("\n Reverse of given Integer %d is %d.",num,reverse);
        return 0;
}
```

Q29. factorial using recursion.

```
#include<stdio.h>
int factorial(int);
int main()
{
 int i=5;
 printf("factorial is %d:",factorial(i));
 return 0;
}
int factorial(int i)
{
 if(i==1)
 {
    return 1;
 }
 return i * factorial(i-1);
}
```

Q30. square of a number.

```
#include<stdio.h>

int main()
{
        int num,square;
        printf("Enter a number : ");
        scanf("%d",&num);
        square=num*num;
        printf("\nThe sqaure of %d is %d",num,square);
        return 0;
}
```

Q31. structure.

```
#include<stdio.h>

struct employee{
        int empid;
        char name[50], empdept[50];
        float salary;
};

int main()
{
        struct employee E;
        printf("\nEnter Employee Id : ");
        scanf("%d",&E.empid);
        printf("Ent Employee Name : ");
    scanf("\n%[^\n]%*c",&E.name);
    printf("Enter Employee Department : ");
    scanf("\n%[^\n]%*c",&E.empdept);
    printf("Enter Employee salary : ");
    scanf("%f",&E.salary);
    printf("\n\nEmployee Id : %d\nEmployee Name : %s\nEmployee Department : %s\nEmployee Salary : %f\n",
    E.empid,E.name,E.empdept,E.salary);
        return 0;
}
```

Q32. subtract matrix.

```
#include<stdio.h>
int main()
{
 int m,w,n,c,d,f[10][10],s[10][10],sub[10][10];
 printf("enter no. of rows and coloumns");
 scanf("%d%d",&m,&n);
 printf("enter elements of first matrix");
 for(c=0;c<m;c++)
 {
   for(d=0;d<n;d++)
    {
      scanf("%d",&f[c][d]);
    }
 }
 printf("enter elements of second matrix");
 for(c=0;c<m;c++)
 {  for(d=0;d<n;d++)
     {
       scanf("%d",&s[c][d]);
     }
 }
```

```
printf("to  perform subtraction of  first matrix form second matrix enter 1,otherwise subtraction of second matrix
scanf("%d",&w);

printf("result after subtraction:");
if(w==1)
  {for(c=0;c<m;c++)
     {
        for(d=0;d<n;d++)
          {
               sub[c][d]=f[c][d]-s[c][d];
          }
     }
    }
else
   { for(c=0;c<m;c++)
      {
        for(d=0;d<n;d++)
          {
             sub[c][d]=s[c][d]-f[c][d];
             printf("%d\t",sub[c][d]);
          }
         printf("\n");
       }
   }
return 0;
}
```

Q33.sum of two numbers.

```
#include<stdio.h>

int main()
{
        int num1,num2,sum;
        printf("Enter the first number : ");
        scanf("%d",&num1);
        printf("\nEnter the second number : ");
        scanf("%d",&num2);
        sum=num1+num2;
        printf("\nSum of the two number is %d\n",sum);
        return 0;
}
```

Q34.sum of pointers.

```
#include <stdio.h>

int main()
{
   int first, second, *p, *q, sum;

   printf("Enter two integers to add\n");
   scanf("%d%d", &first, &second);

   p = &first;
   q = &second;

   sum = *p + *q;

   printf("Sum of the numbers = %d\n", sum);
```

```
    return 0;
}
```

Q35.swaping program.

```
#include <stdio.h>
int main()
{

        int x, y;
        printf("Enter Two Integers : ");
        scanf("%d %d",&x,&y);

        x = x + y;
        y = x - y;
        x = x - y;

        printf("After Swapping: x = %d, y = %d", x, y);

        return 0;
}
```

Q36. table 5.

```
#include<stdio.h>

void main()
{
  int a[10][10],b[10][10];
  int n,m,i,j;
  printf("enter size of matrix as m and n");
  scanf("%d%d",&m,&n);
  printf("\n enter elements of matrix a row wise \n ",m,n);
  for(i=0;i<m;i++)
  { for(j=0;j<n;j++)
    {
       scanf("%d",&a[i][j]);
    }

  }
  for(i=0;i<m;i++)
  { for(j=0;j<n;j++)
     {
       b[j][i]=a[i][j];
     }
  }
  printf("/n transpose is/n/n");

  for(i=0;i<m;i++)
  { for(j=0;j<n;j++)
    {
      printf("%d",b[i][j]);
    }
    printf("\n");
  }
}                                                    return 0;
}
```

Q37. temperature convert.

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

```
int main()
{
        float C,F;
        int choice;
        puts("what do you want to do ? ");
        puts("1. Convert Celsius to Fahrenheit.");
        puts("2. Convert Fahrenheit to Celsius.");
        puts("Enter your choice : ");
        scanf("%d",&choice);
        switch(choice)
        {
                case 1 : printf("Enter Celsius : ");
                scanf("%f",&C);
                F=(C*1.8)+32;
                printf("\n Fahrenheit : %.3f",F);
                break;
                case 2 : printf("Enter Fahrenheit : ");
                scanf("%f",&F);
                C=(F-32)/1.8;
                printf("\n  : %.3f",C);
                break;
        }
        return 0;
}
```

Q38. week days.

```
#include<stdio.h>

int main()

{
        int week;
        printf("\nEnter Week Number : ");
        scanf("%d",&week);
        switch(week){
                case 1 : printf("\nMonday.");
                break;
                case 2 : printf("\nTuesday");
                break;
                case 3 : printf("\nWednesday");
                break;
                case 4 : printf("\nThursday");
                break;
                case 5 : printf("\nFriday");
                break;
                case 6 : printf("\nSaturday");
                break;
                case 7 : printf("\nSunday");
                break;
                default : printf("\nYou have entered wrong week number!");
                break;
        }
        return 0;
}
```

Q39.prime number.

```
#include<stdio.h>

int main()
{
  int i,n,flag=0;
  printf("enter a positive integer:");
```

```
  scanf("%d",&n);
  for(i=2;i<=n/2;++i)
  {
   if(n%i==0)
    {
      flag=1;
      break;
    }
  }
 if (n==1)
 {
   printf(" 1 is neither a prime nor a composite number.");
 }
 else
 {
  if (flag==0)
    printf("%d is a prime number.",n);
  else
    printf("%d is not a prime no.",n);
 }
return 0;
}
```

Q40.quadratic equation.

```
               #include <stdio.h>
#include <math.h>
int main()
{
    double a, b, c, discriminant, root1, root2, realPart, imaginaryPart;
    printf("Enter coefficients a, b and c: ");
    scanf("%lf %lf %lf",&a, &b, &c);
    discriminant = b*b-4*a*c;
    // condition for real and different roots
    if (discriminant > 0)
    {
    // sqrt() function returns square root
        root1 = (-b+sqrt(discriminant))/(2*a);
        root2 = (-b-sqrt(discriminant))/(2*a);
        printf("root1 = %.2lf and root2 = %.2lf",root1 , root2);
    }
    //condition for real and equal roots
    else if (discriminant == 0)
    {
        root1 = root2 = -b/(2*a);
        printf("root1 = root2 = %.2lf;", root1);
    }
    // if roots are not real
    else
    {
        realPart = -b/(2*a);
        imaginaryPart = sqrt(-discriminant)/(2*a);
        printf("root1 = %.2lf+%.2lfi and root2 = %.2f-%.2fi", realPart, imaginaryPart, realPart, imaginaryPart);
    }
    return 0;
}
```

Q41. fizz buzz.

```
#include<stdio.h>
int main()
{
int n;
for(n=1;n<=30;n++)
```

```
{
if(n%3==0 && n%5==0)
printf("Fizzbuzz\n");
else if(n%3==0)
printf("Fizz\n");
else if(n%5==0)
printf(" Buzz\n");
else
printf("\n %d \n",n);
}
return 0;
}
```