13-06-2024

strings using Pointers:-   COPY A STRING

```
-> main ( )
  {
   char a[20], b[20], *p, *q;
   int i;
   p= a; q = b;
   printf ("enter a string ");
   scanf ("%s", a);
   for (i=0; *(p+i)!='\0'; i++)
        *(q+i) = *(p+i);          // *(p+i) = p(i);

   *(q+i) = *(p+i);
   printf ("%s", q);
  }
```

-> String Palindrome /Not using POINTERS.

-> Addition of Matrix using POINTERS

Printing a MATRIX Using POINTERS :-

```
-> main ( )
{
    int a[5][5], r, c, i, j;
    printf (" inter r & c of matrix a ");
    Scanf (" %d %d", &r, &c);
    printf (" enter matrix a");
    for (i = 0; i < r; i++)
    for (j = 0; j < c; j++)
    scanf ("%d", *(a+i)+j);
    for (i = 0; i < r; i++, printf ("\n"))
    for (j = 0; j < c; j++)
    printf ("%d", *(*(a+i)+j));
}
```

Swapping using POINTERS :-

Call by value

call by reference

```
-> void swap (int *, int *);
main ( )
{
    int a = 5, b = 10;
    printf ("Value before swap a = %d, b = %d \n", a, b);
    swap (&a, &b);
    printf ("Values after swap a = %d; b = %d", a, b);
}
void swap (int *a, int *b)
{ int c;
    c = *a; *a = *b; *b = c;
}
```

Area & Perimeter of rectangle using POINTERS :-

```
-> void area ( float, float, float *, float * );
main ( )
{
   float l, b, a, p;
   printf ("enter l, b");
   scanf ("%f %f", &l, &b);
   area (l, b, &a, &p);
      printf ("a = %f, p = %f", a, p);
}
void area (float l, float b, float * a, float *p)
{
   * a = l * b;
   * p = 2 * (l + b);
}
```

Area & circumference of a circle using Pointers :-

Area = πr^2

circumference = $2\pi r$

```
-> void area ( float, float *, float *);
main ()
{
  float r, a, c;
  printf ("enter r :");
  scanf ("%f", &r);
  area (r, &a, &c);
     printf ("a = %f, c = %f", a, c);
}
void area ( float r, float*a, float*c)
{
   *a = 3.14 * r * r;
   *c = 2 * 3.14 * r;
}
```

```
a[i]     *(a+i)

 a[i][j]

  *(a[i]+j)
```

`*(*(a+i)+j)` -> Value at i^{th} row j^{th} column

`*(a+i)+j` -> Pointer to i^{th} row j^{th} column

Palindrome of a String Using POINTERS* :

```
-> Void main ( )
  {
     char str[20], choice;
     int length, flag, i;
     char *p, *q;

     do {
        flag = 0;
        length = 0;
        printf (" Enter a string : ");
        scanf (" %s", str);
        p = str;
        while (*(p + length) ! = '\0')      // Calculate length
                                                of String using
          length ++;                            Pointer

        p = str;
        q = str + length - 1;
        for (i = 0; i < length / 2; i++)      // check string is
                                                Palindrome
        {
           if (*(p+i) ! = *(q-i))            // compare character
                                                using Pointers
           {
              flag = 1;                      // flag if characters don't match
              break;
           }
        }
        if (flag)
           printf ("%s is not a Palindrome", str);
        else
        printf ("%s is a Palindrome ", str);

        printf ("\n\n Do you want to check another string ?
                (y/n): ");
        scanf (" %c", &choice);
      }
      while (choice == 'y' || choice == 'Y');

      printf ("\n\n THANK YOU :) for running the code \n\n");
      return 0;
   }
```

Addition of Matrix Using Pointers* :-

⇒ DOUBLE Dimensional [OR] 2-D Array

```
-> int main()
   {
     int a[100][100], b[100][100], c[100][100], *p,
       *q, *r;
     int i, j, r1, c1;

     printf("Enter no. of rows (between 1 and 100): ");
     scanf("%d", &r1);
     printf("Enter no of columns (between 1 and 100): ");
     scanf("%d", &c1);

     printf("\n Enter matrix A: \n");
      for(i=0; i<r1; i++)
         for(j=0; j<c1; j++)
            scanf("%d", (a[i]+j));

      printf("\n Enter matrix B: \n");
      for(i=0; i<r1; i++)
          for(j=0; j<c1; j++)
             scanf("%d", (b[i]+j));

     p = &a[0][0];
     q = &b[0][0];
     r = &c[0][0];
for (i=0; i<r1; i++)
   for(j=0; j<c1; j++)
      *(r+i*c1+j) = *(p+i*c1+j) +
                    *(q+i*c1+j);

   // Accessing & summing elements at (i, j) & storing
in matrix c

   printf("\n Sum of 2 matrices: \n");
   for(i=0; i<r1; i++)
      for(j=0; j<c1; j++)
      {
         printf("%d ", *(r+i*c1+j));
         if (j==c1-1)
         {
            printf("\n");
         }
      }
}
```