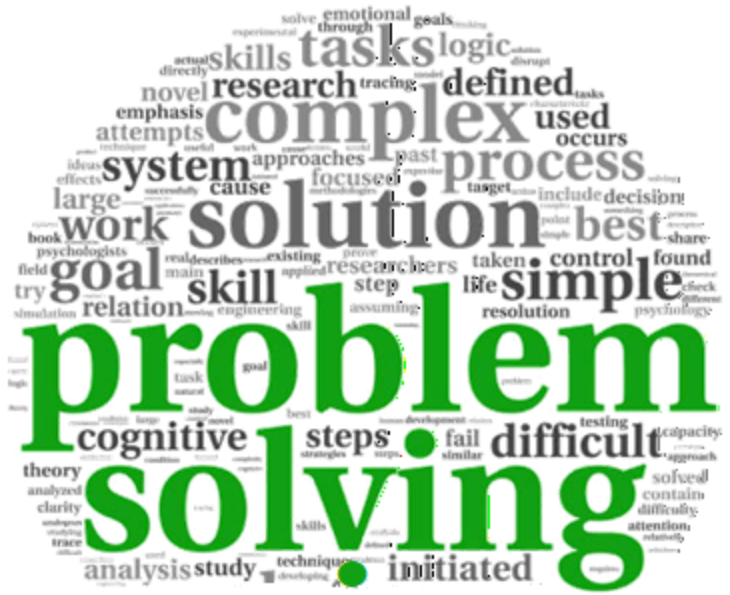


CSE 1051

19.1 PASSING 1-D AND 2-D ARRAY TO FUNCTIONS

Objectives:

To learn and appreciate the following concepts

- Parameter passing techniques
- Passing 1D & 2D arrays to function
- Programs using array & function

Session outcome

At the end of session one will be able to understand:

- The concept of passing arrays to functions
- Understanding problems using arrays and functions

Functions – a recap

```
// FUNCTION DEFINITION
void dispChar( int n, char c) {
    printf(" You have entered %d & %c“,n,c);
 }


int main(){   //calling program
    int no; char ch;
    printf(“Enter a number & a character: \n“);
    scanf(“%d %c”,&no,&ch);
    dispChar( no, ch);   // FUNCTION CALL
    return 0;
}
```

Formal parameters

Actual parameters

Passing 1D-Array to Function

Rules to pass an array to a function

- ✓ The function must be called by passing only the name of the array.
- ✓ In the function definition, the formal parameter must be an array type; the size of the array does not need to be specified.
- ✓ The function prototype must show that argument is an array.

Passing 1D-Array to Function

```
int fnSum( int a[ ], int n) {
    int sum=0,i;
     for(i=0;i<n;i++)
         sum+=a[i];
     return (sum);     }
```

```
int main() {
 int n, a[20], x, y,i;
 printf("Enter the limit \n");
 scanf("%d",&n);
 printf("Enter the values: \n");
 for (i=0; i<n; i++)
 scanf("%d", &a[i]);
 printf("The sum of array elements is =%d ",fnSum(a, n));//fn call
return 0; }
```

Array name is passed along with number of elements

Output: n=5
1, 2, 3, 4, 5
Sum of elements = 15

Passing 2D-Array to Function

Rules to pass a 2D- array to a function

- ✓ The function must be called by passing only the array name.
- ✓ In the function definition, we must indicate that the array has two-dimensions by including two set of brackets.
- ✓ The size of the second dimension must be specified.
- ✓ The prototype declaration should be similar to function header.

Passing 2D-Array to Function

```
int fn2d(int x[ ][10], int m, int n)
  {
       int i, j, sum=0;
       for(i=0; i<m; i++)
           for(j=0; j<n; j++)
               sum+=x[i][j];
       return(sum);
   }
```

```
int main() {
int i, j, a[10][10], m, n;
printf("Enter dimensions of matrix");
scanf("%d%d", &m, &n);
printf("Enter the elements");
for(i=0;i<m;i++)
  for(j=0;j<n;j++)
     scanf("%d",&a[i][j]);
printf ("Sum of elements of 2D array is=%d",fn2d(a, m, n));
return 0;
}
```

Output: m=2 n=3

1 2

3 4

5 6

Sum of elements = 21

Write a c program to add all the even elements of an array using a function Add()

```
int main(){
  int n, a[20], x, y,i;
  printf("Enter the limit \n");
  scanf("%d",&n);
  printf("Enter the values: \n");
  for (i=0; i<n; i++)
    scanf("%d",&a[i]);
  printf("sum of even elements is =%d ",Add(a,n));
 return 0;
}
```

```
int Add( int a[ ], int n)
 {
    int sum=0,i;
    for(i=0;i<n;i++)
    {
        if((a[i]%2) == 0)
            sum+=a[i];
    }
    return (sum);
 }
```

Output: n=5
1 2 3 4 5
Sum of even elements = 6

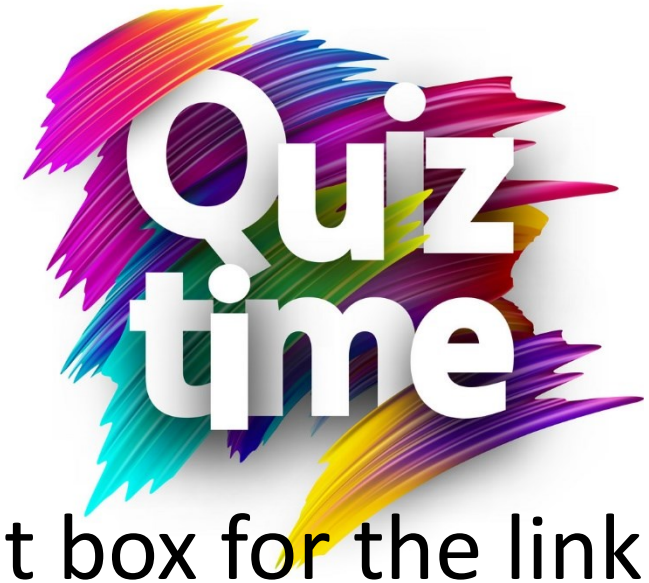
Write a C program to replace all odd numbers of an array with the largest number in the array using a function Replace()

```
void Replace( int arr[ ], int n)
{
    // To find the largest
     int i, max = arr[0];
     for (i = 1; i < n; i++)
         if (arr[i] > max)
             max = arr[i];

    // To replace
     for(i=0;i<n;i++)
     {
         if(arr[i]%2 != 0)
             arr[i]=max;
     }
 }
```

```
int main()  {
   int n, a[20], x, y,i;
   printf("Enter the limit \n");
   scanf("%d",&n);
   printf("Enter the values: \n");
   for (i=0; i<n; i++)
        scanf("%d",&a[i]);
 Replace(a,n);
 printf("The array after replacement is\n");
 for (i=0; i<n; i++)
   printf("%d \n",a[i]);

return 0;
}
```



Go to posts/chat box for the link to the question

submit your solution in next 2 minutes

The session will resume in 3 minutes

Write a C program to replace all the zeros in the matrix by the trace of a square matrix using a function Trace()

```
void Trace(float a[ ][10],  int n)
{
  int i, j, tr=0;

  // Finding Trace
  for(i=0;i<n;i++)
  {
       tr= tr + a[ i ][ i ];
  }
//Replacing zeros
  for(i=0;i<n;i++)
     for(j=0;j<n;j++)
        if(a[i][j]==0)
            a[i][j]=tr;
}
```

```
int main() {
 int i, j, n;
 float a[10][10];
 printf("Enter the rows or columns of  a square matrix");
 scanf("%d", &n);
 printf("Enter the elements");
 for(i=0;i<n;i++)
   for(j=0;j<n;j++)
      scanf("%f",&a[i][j]);
 Trace(a, n);
 printf("Matrix after replacement \n");
 for(i=0;i<n;i++)   {
    for(j=0;j<n;j++)     {
       printf("%f",a[i][j]);
    }
    printf("\n");
 }
 return 0;
}
```

Write a C program using pass-by-pointer method to compute the simple interest and compound interest using a function SI_CI()

```
void SI_CI(float *pr, float *ra, int *yr,
float *si, float *ci)
{
    int amount;

    // Simple interest
    *si = ((*pr)*(*ra)*(*yr)/100);

    // Compound interest
    amount= (*pr)*pow((1 +(*ra)/100),(*yr));
        *ci= amount-(*pr);
}
```

```
#include <math.h>
int main() {
 float p,q,r,SI,CI;
  int n;
  printf("Enter the value of Principal p = ");
  scanf("%f",&p);
  printf("Enter the value of Rate r = ");
  scanf("%f",&r);
  printf("Enter the value of Period in year n = ");
  scanf("%d",&n);

  SI_CI( &p, &r, &n, &SI, &CI );

  printf("Simple Interest SI=%f \n", SI);
  printf("Compound Interest CI=%f \n", CI);
 return 0;
}
```

Summary:

- Parameter passing techniques
  - pass by value

    void swap(int x, int y)
  - pass by reference

    void swap(int *x, int *y)
- Passing 1 D array

  int fnParr(int a[ ], int n)
- Passing 2 D array

  int fn2d(int x[ ][10], int m, int n)