

Q1 a) What is an Array? Explain declaration and initialization for 2-D array with examples. 5 M

Ans:

An array is an identifier that refers to the collection of data items which all have the same name. The individual data items are represented by their corresponding array elements. Address of an element (subscript) ranges from 0 to n-1, where n is total number of elements.

Declaration of Two-dimensional array:

syntax:

```
type arrayname[rowsize][columnsize];
```

Examples:

```
int stud[16][5];
```

Initialization of Two-dimensional array:

Examples:

```
int marks[3][3] = {  
                    {26,57,66},  
                    {56,77,48},  
                    {76,54,82}  
                    };
```

```
int disp[2][4] = {  
    {10, 20, 30, 13},  
    {-17, 0, 40, 17}  
};
```

Or:

```
int disp[2][4] = { 10, 20, 30, 13, -17, 0, 40, 17};
```

If any element is initialized, by default rest of the elements are initialized to 0.

```
int a[5][4] = {{0}}; /* first element is 0 & rest all elements also 0 */
```

```
int marks[3][3] = { {12},{10},{20}};
```

Here first element of each row is explicitly initialized to 12,

2nd row first element i.e. marks[1][0] is 10

3rd row first element i.e. marks[2][0] is 20

while other elements are automatically initialized to zero.

Q1b) Write a program to find largest element in N elements of integer array

5 Marks

Ans:

```
/* largest in n elements array*/  
#include <stdio.h>  
int main()  
{  
    int a[1000],x,n,max;  
    printf("Number of elements? ");  
    scanf("%d",&n);  
    for(x=0;x<n;x++)  
    {  
        printf("Enter element %d : ", x+1);  
        scanf("%d",&a[x]);  
        if(x==0) max=a[0]; /* first element assumed as largest */  
    }  
}
```

```

        if(a[x] > max) max=a[x]; /*comparing rest to find largest */
    }
    printf("The largest element is %d\n",max);

    return (0);
}

```

Output:

Number of elements? 11

Enter element 1 : 33

Enter element 2 : 4

Enter element 3 : 55

Enter element 4 : 6

Enter element 5 : 77

Enter element 6 : 8

Enter element 7 : 76

Enter element 8 : 54

Enter element 9 : 32

Enter element 10 : 43

Enter element 11 : 12

The largest element is 77

Q 2. Write a C program for Multiplication of 2 matrices.

10 M

Ans:

```

/* Multiplication of matrices */
#include<stdio.h>
int main()
{
    int a[100][100],b[100][100],c[100][100],i,j,k,m,n,p;
    printf("Enter total rows and cols of matrix a : ");
    scanf("%d%d",&m,&n);
    printf("Total column of matrix b : ");
    scanf("%d",&p);

    printf("Enter matrix a - %d x %d:\n",m,n);
    for(i=0;i<m;i++)
        for(j=0;j<n;j++)
            scanf("%d",&a[i][j]);

    printf("Enter matrix b - %d x %d :\n",n,p);
    for(i=0;i<n;i++)
        for(j=0;j<p;j++)
            scanf("%d",&b[i][j]);

    /* Multiplying & displaying*/
    printf("Resultant Matrix:\n");
    for(i=0;i<m;i++)
    {
        for(j=0;j<p;j++)
        {
            c[i][j]=0;
            for(k=0;k<n;k++)
                c[i][j] += a[i][k] * b[k][j];
            printf("%5d",c[i][j]);
        }
        printf("\n");
    }
}

```

```

    }

return (0);
}

```

Output:

Enter total rows and cols of matrix a : 4 3

Total column of matrix b : 4

Enter matrix a - 4 x 3:

2 3 4

3 2 4

2 1 3

3 2 2

Enter matrix b - 3 x 4 :

2 3 4 1

2 3 4 3

3 2 3 2

Resultant Matrix:

22 23 32 19

22 23 32 17

15 15 21 11

16 19 26 13

Q3. What is Binary search? Implement Binary search Program.

10 M

Ans:

```

/* Binary search in given array of N elements */
#include <stdio.h>

int main()
{
    int a[1000],x, y,temp,N, first, last, mid, skey, found;
    printf("Enter number of elements : ");
    scanf("%d", &N);
    for(x=0;x<N;x++)
    {
        printf("Enter element %d : ", x+1);
        scanf("%d",&a[x]);
    }
    /* now sorting - because Binary search requires sorted elements */
    for(x=0;x<N;x++)
        for(y=0;y<N-x-1;y++)
            if(a[y] > a[y+1])
            {
                temp=a[y];
                a[y]=a[y+1];
                a[y+1]=temp;
            }
    printf("The sorted array:\n");
    for(x=0;x<N;x++)
        printf(" %d",a[x]);

    printf("\nEnter search key integer : ");
    scanf("%d",&skey);
    first=0;
    last=N-1;
    found=0;

    while(first <= last && ! found)

```

```

{
    mid=(first+last)/2; /* integer value of mid */
    if(a[mid] > skey) last=mid-1;
    else if (a[mid] < skey)first=mid+1;
    else found=1;
}
if(found)
    printf("Found at %dth position\n",mid+1);
else
    printf("Not found\n");
return (0);
}

```

Expected Ouput:

```

Enter number of elements : 11
Enter element 1 : 22
Enter element 2 : 3
Enter element 3 : 44
Enter element 4 : 5
Enter element 5 : 66
Enter element 6 : 7
Enter element 7 : 88
Enter element 8 : 9
Enter element 9 : 111
Enter element 10 : 23
Enter element 11 : 54
The sorted array:
 3 5 7 9 22 23 44 54 66 88 111
Enter search key integer : 88
Found at 10th position

```

Q 4. Explain any 5 string library functions with examples.

10 M

Ans:

Header file for all string handling functions is **string.h**

i. **strlen(str) – Returns length of string (in bytes – Number of characters in str)**

Syntax:

```
int strlen(const char str[]);
```

Example:

```
printf("Length of Dr. P. N. Singh string is %d\n",strlen("Dr. P. N. Singh"));
```

Output: Length of Dr. P. N. Singh string is 15

ii. **strrev(str) – reverses the string**

Syntax:

```
Char *strrev(char *);
```

Example:

```
printf("Reverse of \"Listen\" is %s\n",strrev("Listen"));
```

Output: "Reverse of "Listen" is netsiL

iii. **strcpy(dest_str, src_str) – Copies string source o destination**

Syntax:

```
char *strcpy(char *dest, const char *scr);
```

Example:

```
char name[ 5];
```

```
strcpy(name, "Paras");
printf("name is %s\n", name);
Output: name is Paras
```

iv. strcat(str1, str2) – Concatenates(append) str2 to str1 at the end.

Syntax:

```
char* strcat (char* strg1, const char* strg2);
```

Example:

If two strings are:

```
char name[]="Roma";
char surname[]=" Ritambhara";
```

Then to append "Ritambhara" with name "Roma"

v. strcat(name,surname);

```
printf("%s", name);
```

Output: Roma Ritambhara

strcmp(str1, str2) – Compares two string and returns:

- 0 if both strings are identical
- else returns difference based on ASCII value of first mismatch(1 or -1).

Syntax:

```
int strcmp (const char* str1, const char* str2);
```

Example:

```
printf("%d", strcmp("SINGH", "SINHA");
```

output: -1

Q 5. Write a program to check a string whether it is palindrome.

10 M

Ans:

```
/* To check a string(with space) whether it is palindrome
without using built-in function*/
```

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

```
int main()
```

```
{
```

```
char str[50];
```

```
int x,y,len=0,palin=1;
```

```
printf("Enter string : ");
```

```
while((str[len]=getchar())!='\n')
```

```
len++;
```

```
str[len]='\0'; /* last character NULL */
```

```
y=len-1; /* coming before last character */
```

```
for(x=0;str[x]!='\0';x++,y--)
```

```
if(str[x]!=str[y])
```

```
{
```

```
palin=0;
```

```
break;
```

```
}
```

```
if(palin)
```

```
printf("Yes, \"%s\" is a palindrom\n",str);
```

```

else
    printf("No, \"%s\" is not a palindrome\n",str);

return (0);
}

```

Output: Enter string: able was I ere I saw elba
Yes, able was I ere I saw elba is a palindrome

Q 6. What is user defined function? Explain the different categories of User defined function?

10M

Ans:

Here we discuss about user defined function.

- A function is a self-contained block of statements that performs a coherent task of some kind. Arguments are written within () parentheses separating by comma ,
- A function may or may not have arguments.
- A function may or may not return a value.
- A function can return one and only one value.

Function declaration:

Formal declaration of a function with its return type and types of arguments are known as prototype.

Syntax for declaring a function:

returntype functionname(type1, type2, type3,...);

example:

int rectarea(int, int);

Function Definition:

Operations are performed in function definition for which task is related. Arguments (parameters) are received from calling function with their types.

```

int rectarea(int len, int wid)
{
    int a = len*wid;
    return (a);
}

```

Function Call:

Functions are called passing the arguments only. Arguments should be as type of arguments are declared in function.

```

rectarea(x,y);
/* Now bringing everything together */
#include <stdio.h>
int rectarea(int, int); /* function declaration */
int main()
{
    int x,y,area;
    printf("Enter length & width : ");
    scanf("%d%d", &x, &y);
    area=rectarea(x,y); /* function call */
    printf("Area of rectangle = %d square unit\n", area);
    return (0);
}

int rectarea(int len, int wid)
{
    int a = len*wid;
    return (a);
}

```

Categories of user defined function:

- i. function without argument/parameter
- ii. function with parameters
- iii. function returning a value
- iv. void functions/functions does not return a value

Again it can be correlated

- i. functions without parameters & not returning any value

```
void gao() { printf("Kudi punjaban dil chura ke le gayee sona sona, dil mera sona"); }
```
- ii. functions without parameters returning a value

```
int boolnum() { /*...*/ return (1); }
```
- iii. function with parameter returns a value:

```
int rectarea(int a, int b) { return (a*b); }
```
- iv. function with parameter does not return a value:

```
void rectarea(int a, int b) { printf("Area = %d\n",a*b); }
```

Q 7. Demonstrate Bubble sort for sorting numbers 22, 3, 44, 5, 66, 7, 77, 8, 99 & 9 for each pass. 10 M

Ans:
 In Bubble sort technique adjacent elements are compared. In each pass "heaviest" items sinks down and "lighter" items bubble up. Hence, the technique is now as "Bubble" sort.

Demo:

| | | | | | | | | |
|----|----|----|----|----|----|----|----|----|
| 22 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| 3 | 22 | 22 | 22 | 22 | 22 | 22 | 22 | 22 |
| 44 | 44 | 44 | 5 | 5 | 5 | 5 | 5 | 5 |
| 5 | 5 | 5 | 44 | 44 | 44 | 44 | 44 | 44 |
| 66 | 66 | 66 | 66 | 66 | 7 | 7 | 7 | 7 |
| 7 | 7 | 7 | 7 | 7 | 66 | 66 | 66 | 66 |
| 77 | 77 | 77 | 77 | 77 | 77 | 77 | 8 | 8 |
| 8 | 8 | 8 | 8 | 8 | 8 | 8 | 77 | 77 |
| 99 | 99 | 99 | 99 | 99 | 99 | 99 | 99 | 99 |
| 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 |

First Pass

| | | | | | | | |
|----|----|----|----|----|----|----|----|
| 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| 22 | 22 | 5 | 5 | 5 | 5 | 5 | 5 |
| 5 | 5 | 22 | 22 | 22 | 22 | 22 | 22 |
| 44 | 44 | 44 | 44 | 7 | 7 | 7 | 7 |
| 7 | 7 | 7 | 7 | 44 | 44 | 44 | 44 |
| 66 | 66 | 66 | 66 | 66 | 66 | 8 | 8 |
| 8 | 8 | 8 | 8 | 8 | 8 | 66 | 66 |
| 77 | 77 | 77 | 77 | 77 | 77 | 77 | 77 |
| 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 |
| 99 | 99 | 99 | 99 | 99 | 99 | 99 | 99 |

Second Pass

| | | | | | | |
|----|----|----|----|----|----|----|
| 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| 22 | 22 | 22 | 7 | 7 | 7 | 7 |
| 7 | 7 | 7 | 22 | 22 | 22 | 22 |
| 44 | 44 | 44 | 44 | 44 | 8 | 8 |
| 8 | 8 | 8 | 8 | 8 | 44 | 44 |
| 66 | 66 | 66 | 66 | 66 | 66 | 66 |
| 9 | 9 | 9 | 9 | 9 | 9 | 9 |
| 77 | 77 | 77 | 77 | 77 | 77 | 77 |
| 99 | 99 | 99 | 99 | 99 | 99 | 99 |

Third Pass

| | | | | | |
|----|----|----|----|----|----|
| 3 | 3 | 3 | 3 | 3 | 3 |
| 5 | 5 | 5 | 5 | 5 | 5 |
| 7 | 7 | 7 | 7 | 7 | 7 |
| 22 | 22 | 22 | 22 | 8 | 8 |
| 8 | 8 | 8 | 8 | 22 | 22 |
| 44 | 44 | 44 | 44 | 44 | 44 |
| 9 | 9 | 9 | 9 | 9 | 9 |
| 66 | 66 | 66 | 66 | 66 | 66 |
| 77 | 77 | 77 | 77 | 77 | 77 |
| 99 | 99 | 99 | 99 | 99 | 99 |

4th Pass

| | | | | |
|----|----|----|----|----|
| 3 | 3 | 3 | 3 | 3 |
| 5 | 5 | 5 | 5 | 5 |
| 7 | 7 | 7 | 7 | 7 |
| 8 | 8 | 8 | 8 | 8 |
| 22 | 22 | 22 | 22 | 22 |
| 9 | 9 | 9 | 9 | 9 |
| 44 | 44 | 44 | 44 | 44 |
| 66 | 66 | 66 | 66 | 66 |
| 77 | 77 | 77 | 77 | 77 |
| 99 | 99 | 99 | 99 | 99 |

5th Pass

| | | | | | |
|--------------|----|----|----|--|--|
| 3 | 3 | 3 | 3 | | |
| 5 | 5 | 5 | 5 | | |
| 7 | 7 | 7 | 7 | | |
| 8 | 8 | 8 | 8 | | |
| 9 | 9 | 9 | 9 | | |
| 22 | 22 | 22 | 22 | | |
| 44 | 44 | 44 | 44 | | |
| 66 | 66 | 66 | 66 | | |
| 77 | 77 | 77 | 77 | | |
| 99 | 99 | 99 | 99 | | |
| 6th Pass | | | | | |
| 3 | 3 | 3 | | | |
| 5 | 5 | 5 | | | |
| 7 | 7 | 7 | | | |
| 8 | 8 | 8 | | | |
| 9 | 9 | 9 | | | |
| 22 | 22 | 22 | | | |
| 44 | 44 | 44 | | | |
| 66 | 66 | 66 | | | |
| 77 | 77 | 77 | | | |
| 99 | 99 | 99 | | | |
| 7th Pass | | | | | |
| 3 | 3 | | | | |
| 5 | 5 | | | | |
| 7 | 7 | | | | |
| 8 | 8 | | | | |
| 9 | 9 | | | | |
| 22 | 22 | | | | |
| 44 | 44 | | | | |
| 66 | 66 | | | | |
| 77 | 77 | | | | |
| 99 | 99 | | | | |
| 8th Pass | | | | | |
| 3 | 3 | | | | |
| 5 | 5 | | | | |
| 7 | 7 | | | | |
| 8 | 8 | | | | |
| 9 | 9 | | | | |
| 22 | 22 | | | | |
| 44 | 44 | | | | |
| 66 | 66 | | | | |
| 77 | 77 | | | | |
| 99 | 99 | | | | |
| 9th Pass | | | | | |
| 3 | | | | | |
| 5 | | | | | |
| 7 | | | | | |
| 8 | | | | | |
| 9 | | | | | |
| 22 | | | | | |
| 44 | | | | | |
| 66 | | | | | |
| 77 | | | | | |
| 99 | | | | | |
| sorted array | | | | | |

Total number of comparisons: $n*(n-1)/2$

Q 8. What is selection sort? Write a program to implement selection sort using function. 10 M

Ans:

```

/* selection sort using function */
#include <stdio.h>
void selesort(int a[], int n)
{
    int x,y,temp;
    for(x=0;x<n-1;x++)
        for(y=x+1;y<n;y++)/* elements are compared to select minimum*/
            if(a[x] > a[y])
                {
                    temp=a[x];        /*swapping */
                    a[x]=a[y];
                    a[y]=temp;
                }
}

int main()
{
    int a[1000],x,n;
    printf("Number of elements? ");
    scanf("%d",&n);
    for(x=0;x<n;x++)
        {
            printf("Enter element %d : ", x+1);
            scanf("%d",&a[x]);
        }
    selesort(a,n); /* calling selesort() function */
    printf("Sorted array\n");
    for(x=0;x<n;x++)
        printf(" %d",a[x]);
    return (0);
}

```

Output:

Number of elements? 9
Enter element 1 : 22
Enter element 2 : 3

Enter element 3 : 44

Enter element 4 : 5

Enter element 5 : 66

Enter element 6 : 7

Enter element 7 : 77

Enter element 8 : 8

Enter element 9 : 9

Sorted array

3 5 7 8 9 22 44 66 77