

ptr[0] points to the a[0][0] (0th row-0th column)
 ptr[1] points to the a[1][0] (1st row-0th column)
 ptr[2] points to the a[2][0] (2nd row-0th column)
 ptr[3] points to the a[3][0] (3rd row-0th column)
 ptr[4] points to the a[4][0] (4th row-0th column)

4.12 TYPES OF POINTERS: VOID AND NULL POINTERS

Void Pointers

A void pointer is nothing but a pointer variable declared using the reserved word in C 'void'. For example:

```
void *ptr; // Now ptr is a general purpose pointer variable
```

When a pointer variable is declared using keyword void – it becomes a general purpose pointer variable. Address of any variable of any data type (char, int, float etc.) can be assigned to a void pointer variable.

Dereferencing a void pointer

We can't directly dereference a void pointer; we must cast it to a pointer with a specific type first, for instance, to make it behave like pointer of integer type we have to cast to int*:

```
*(int*)ptr;
```

Thus to assign a value to a void pointer, you will have to do something like:

```
*(int*)ptr=40;  
*(float*)pt=3.14;
```

Example program: - Using void pointers

```
#include<stdio.h>
void main()
{
  int a=10;
  float b=35.75;
  void *ptr; // Declaring a void pointer
  ptr=&a; // Assigning address of integer to void pointer.
  printf("The value of integer variable is= %d",*( (int*) ptr));
  // (int*)ptr - is used for type casting. Where as *( (int*)ptr) dereferences the typecasted void pointer variable.
  ptr=&b; // Assigning address of float to void pointer.
  printf("The value of float variable is= %f",*( (float*) ptr));
}
```

Output:

The value of integer variable is = 10

The value of float variable is = 37.75

Null Pointers

A null pointer in C is a pointer that is assigned to zero or NULL where a variable that has no valid address. The null pointer usually does not point to anything.

Syntax

```
int *pointer_var = NULL;
```

Or We can directly assign the pointer variable to 0 to make it null pointer.

```
int *pointer_var = 0
```

Let us see an example of how null pointers are created.

Example: Using null pointers

```
#include <stdio.h>
int main ()
{
    int *ptr = (int *)NULL;
    printf("The value of pointer assigned is : %x\n", ptr );
    return 0;
}
```

Output:

The value of pointer assigned is : 0

Explanation: In the above code, we are initializing the variable “ptr” to 0 (zero) so when we print the pointer value which Null pointer.

EXERCISE**■ MCQ**

1. C allows arrays of greater than two dimensions, who will determine this
 (a) programmer (b) compiler (c) parameter (d) None of these
Ans. : (b)
2. A pointer to a pointer is a form of
 (a) multiple indirections (b) a chain of pointers (c) both a and b (d) None of these
Ans. : (c)
3. Pointers are of
 (a) integer data type (b) character data type
 (c) unsigned integer data types (d) None of these
Ans. : (d)

4. Maximum number of elements in the array declaration int a[5][8] is

(a) 28 (b) 32 (c) 35 (d) 40

Ans. : (d)

5. If the size of the array is less than the number of initializers then,

(a) Extra values are being ignored (b) Generates an error message
(c) Size of Array is increased (d) Size is neglected when values are given

Ans. : (a)

6. Array subscripts in C always start at

(a) -1 (b) 1 (c) 0 (d) Value provided by the user

Ans. : (c)

7. Which is the correct way to declare a pointer?

(a) int_ptr; (b) int *ptr; (c) *int ptr; (d) None of these.

Ans. : (b)

8. If you want to exchange two rows in a two-dimensional array, the fastest way is to:

(a) Exchange the elements of the 2rows
(b) Exchange the address of each element in the two-row
(c) Silence the address of the rows in an array of the pointer and exchange the pointer
(d) None of these.

Ans. : (c)

■ Questions

1. What is an array? Explain Single Dimensional array with suitable sample program in 'C'.
2. Explain 1D array declaration and initialization giving example.
3. Explain 2D array declaration and initialization giving example.
4. Declare a character-based array called letters of ten elements
5. Use a for loop to total the contents of an integer array called numbers which has five elements. Store the result in an integer called total.
6. What is pointer? Explain with example.
7. What is "Address of(&)" operator? Explain with example.
8. Explain pointer arithmetic with example.
9. Explain pointer to array with example.
10. Explain array of pointers with example.
11. Explain pointer to pointer with example.
12. Explain null pointer with example.

13. Explain advantages and disadvantages of pointers.
14. What is difference between static memory allocation and dynamic memory allocation? Give example.
15. Explain difference between malloc and calloc function.
16. Declare a pointer to an integer called address.
17. Assign the address of a float variable balance to the float pointer temp.
18. Assign the character value ‘W’ to the variable pointed to by the char pointer letter.
19. What is the output of the following program segment?
20. Declare a pointer to the text string “Hello” called message.

■ Programming Exercise

1. Given a two-dimensional array, write a program that totals all elements, printing the total.
2. Write a program to find out of multiplication of two matrixes.
3. Write a program to find out the transpose of a matrix.
4. Write a program to prepare the pay bill for the employees of a company
5. Write a program to delete any element from any array.
6. Write a program to search data from any array
7. Write a C program to read through an array of any type using pointers. Write a C program to scan through this array to find a particular value.
8. Write a program to find the number of times that a given word(i.e. a short string) occurs in a sentence (i.e. a long string!).
9. Write a program that takes three variable (a, b, c) in as separate parameters and rotates the values stored so that value a goes to be, b, to c and c to a.
10. Write a C program to read through an array of any type using pointers. Write a C program to scan through this array to find a particular value.
11. Write a program that reads a number that says how many integer numbers are to be stored in an array, creates an array to fit the exact size of the data and then reads in that many numbers into the array.

GTU Exam Paper Solution

Questions 2 Marks

1. What is the importance of an array in C.

Summer 2019

Ans. Consider a case where you need to find out the average of 100 integer numbers entered by user. In C, you have two ways to do this : (1) Define 100 variables with int data type and then perform 100 scanf() operations to store the entered values in the variables and then at last calculate the average of them. (2) Have a single integer array to store all the values, loop the array to store all the entered values in array and later calculate the average.

2. List out limitations of an array.

Summer 2020

- Ans.**
- The number of elements to be stored in an array should be known in advance.
 - An array is a static structure (which means the array is of fixed size). Once declared the size of the array cannot be changed.
 - Insertion and deletion operations are quite difficult in an array as the elements are stored in consecutive memory locations.
 - Allocating more memory than the requirement leads to wastage of memory space and less allocation of memory also leads to a problem

3. Define an array of the size 3×4 and initialize it.

Summer 2020

Ans.

```
#include <stdio.h>
int main() {
    int r=3, c=3, a[3][4], i, j;

    printf("\nEnter elements of matrix:\n");
    for (i = 0; i < r; ++i)
        for (j = 0; j < c; ++j) {
            printf("Enter element a%d%d: ", i + 1, j + 1);
            scanf("%d", &a[i][j]);
        }

    // printing the matrix
    printf("\nMatrices: \n");
    for (i = 0; i < r; ++i)
        for (j = 0; j < c; ++j) {
            printf("%d ", a[i][j]);
            if (j == c - 1) {
                printf("\n\n");
            }
        }

    return 0;
}
```

4. Define pointer. Give the syntax to initialize it.

Summer 2020

Please refer 4.9

5. Give two methods to input a string with space.

Summer 2020

Ans.

Method 1 : Using gets

Syntax : char *gets(char *str)

```
#include <stdio.h>
int main()
{
    char str[20];
    gets(str);
    printf("%s", str);
    return 0;
}
```

Method 2 : Using %[^n]*c inside scanf

Syntax : scanf("%[^n]*c", str);

6. If declaration is char name[10]; then what is the maximum length of string name?

Summer 2020-2

Ans. 9

7. What do you mean by actual argument?

Summer 2020-2

Ans. The arguments that are passed in a function call are called actual arguments. These arguments are defined in the calling function. These are the variables or expressions referenced in the parameter list of a subprogram call. There is no need to specify datatype in actual parameter.

8. What is Array? Give four Characteristics.

Summer 2021

Ans. Please refer 4.1, 4.2

9. Explain initialization of two-dimension an array.

Summer 2021

Ans. Please refer 4.4.2

Questions 3 Marks

1. Write a C program to calculate the sum of all the elements of an array of 6 integers.

Winter 2019

Ans.

```
#include <conio.h>

int main()
{
    int a[6], i, n = 6, sum = 0;

    printf("Enter elements in array : ");
    for (i = 0; i < n; i++)
    {
        scanf("%d", &a[i]);
    }
```

```

for(i=0; i<n; i++)
{
    sum+=a[i];
}
printf("sum of array is : %d",sum);

return 0;
}

```

2. Write a program to read array often integer numbers and print only even numbers from it.

Winter 2018

Ans.

```

#include<stdio.h>
int main()
{
    int n, a[20];

    printf("Enter the size of the array: ");
    scanf("%d", &n);

    printf("Enter array elements: \n");
    for(int i=0; i<n; i++)
    {
        scanf("%d", &a[i]);
    }

    printf("Even numbers in the array are: \n");
    for(int i=0; i<n; i++)
    {
        if(a[i]%2==0)
            printf("%d ", a[i]);
    }

    return 0;
}

```

3. Write a program to read array of five integer number and calculate average of that number. Summer 2019

Ans.

```

#include <stdio.h>
int main() {
    int n=5, i;
    float num[5], sum = 0.0, avg;
}

```

```

printf("Enter the numbers of elements: ");
scanf("%d", &n);

for (i = 0; i < n; ++i) {
    printf("Enter number: ", i + 1);
    scanf("%d", &num[i]);
    sum += num[i];
}

avg = sum / n;
printf("Average = %.2f", avg);
return 0;
}

```

4. Explain pointer to pointer with example.

Summer 2020-2

Ans. Please refer 4.10

5. Explain void pointer.

Summer 2021

Ans. Please refer 4.12

Questions 4 Marks

1. What is array? How to initialize one dimensional array.

Winter 2018

What is an array? How to initialize one dimensional array?

Summer 2019

Ans. Please refer 4.1, 4.3

2. Write characteristic of Array

Winter 2018

Ans. Please refer 4.2

3. Write a C program to input and add two matrices of 2x3. Display final matrix with proper format.

Ans. Summer 2020

```

#include <stdio.h>
int main() {
    int r=2, c=3, a[2][3], b[2][3], sum[2][3], i, j;
    printf("Enter the number of rows (between 1 and 100): ");

    printf("\nEnter elements of 1st matrix:\n");
    for (i = 0; i < r; ++i)
        for (j = 0; j < c; ++j) {
            printf("Enter element a%d%d: ", i + 1, j + 1);
            scanf("%d", &a[i][j]);
        }
}

```

```

printf("Enter elements of 2nd matrix:\n");
for (i = 0; i < r; ++i)
    for (j = 0; j < c; ++j) {
        printf("Enter element b%d%d: ", i + 1, j + 1);
        scanf("%d", &b[i][j]);
    }

// adding two matrices
for (i = 0; i < r; ++i)
    for (j = 0; j < c; ++j) {
        sum[i][j] = a[i][j] + b[i][j];
    }

// printing the result
printf("\nSum of two matrices: \n");
for (i = 0; i < r; ++i)
    for (j = 0; j < c; ++j) {
        printf("%d ", sum[i][j]);
        if (j == c - 1) {
            printf("\n\n");
        }
    }

return 0;
}

```

4. Explain 'addressof' and indirection operator of pointer with example.

Summer 2020

Ans. Please refer 4.8

5. Discuss array of pointer with example.

Summer 2020-2

Ans. Please refer 4.11.2

6. Write a program to multiply a 3×3 matrix array with integer number scanned from user.

Ans.

```

#include<stdio.h>

int main()
{
    int a[3][3],b[3][3],c[3][3],i,j,k,sum;

    printf("\nEnter the matrix elements of A\n");
    for(i=0;i<3;i++)
    {

```

```

for(j=0;j<3;j++)
{
    scanf("%d",&a[i][j]);
}
printf("\n");
}

printf("\nEnter the matrix elements of B\n");
for(i=0;i<3;i++)
{
    for(j=0;j<3;j++)
    {
        scanf("%d",&b[i][j]);
    }
    printf("\n");
}
for(i=0;i<3;i++)
{
    for(j=0;j<3;j++)
    {
        sum=0;
        for(k=0;k<3;k++)
        {
            sum=sum+a[i][k]*b[k][j];
        }
        c[i][j]=sum;
    }
}
printf("\nProduct of two matrix is:\n\n");
for(i=0;i<3;i++)
{
    for(j=0;j<3;j++)
    {
        printf("%d",c[i][j]);
        printf("\t");
    }
    printf("\n\n");
}
return 0;
}

```

7. Explain array of pointer with example.

Please refer 4.11.2

