11)Define and declare pointer.

What is Pointer?

- A normal variable is used to store value.
- A pointer is a variable that store address / reference of another variable.
- Pointer is derived data type in C language.
- A pointer contains the memory address of that variable as their value.
- Pointers are also called address variables because they contain the addresses of other variables.

2

Declaration & Initialization of Pointer

Syntax

```
datatype *ptr_variablename;
```

Example

```
void main()
{
    int a=10, *p; // assign memory address of
    a to pointer variable p
    p = &a;
    printf("%d %d %d", a, *p, p);
}
```

Output

```
10 10 5000
```

Variable	Value	Address
a	10	5000
p	5000	5048

- p is integer pointer variable
- & is address of or referencing operator which returns memory address of variable.
- * is indirection or dereferencing operator which returns value stored at that memory address.
- & operator is the inverse of * operator
- x = a is same as x = *(&a)

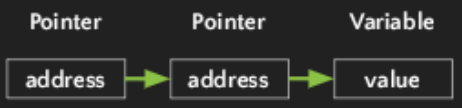
13)Define pointer to pointer. Write a C program which illustrates the use of pointer to pointer

Pointer to Pointer – Double Pointer

- Pointer holds the address of another variable of same type.
- When a pointer holds the address of another pointer then such type of pointer is known as pointer-to-pointer or double pointer.
- The first pointer contains the address of the second pointer, which points to the location that contains the actual value.

Syntax

```
datatype **ptr_variablename;
```



Example

```
int **ptr;
```

7

Write a program to print variable, address of pointer variable and pointer to pointer variable.

Program

```
#include <stdio.h>
int main () {
    int var;
    int *ptr;
    int **pptr;
    var = 3000;
    ptr = &var; // address of var
    pptr = &ptr; // address of ptr using address of operator &
    printf("Value of var = %d\n", var );
    printf("Value available at *ptr = %d\n", *ptr );
    printf("Value available at **pptr = %d\n", **pptr);
    return 0;
}
```

Output

```
Value of var = 3000
Value available at *ptr = 3000
Value available at **pptr = 3000
```

8

16)Describe the concept of pointer to an array with an example.

Array of Pointer

- As we have an array of char, int, float etc, same way we can have an array of pointer.
- Individual elements of an array will store the address values.
- So, an array is a collection of values of similar type. It can also be a collection of references of similar type known by single name.

Syntax

```
datatype *name[size];
```

Example

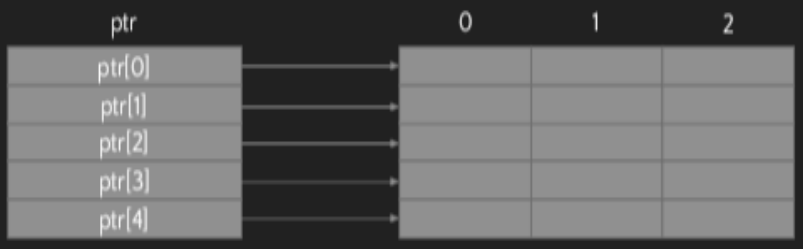
```
int *ptr[5]; //declares an array of integer pointer of size 5
```

Array of Pointer – Cont.

- An array of pointers ptr can be used to point to different rows of matrix as follow:

Example

```
for(i=0; i<5; i++)
{
    ptr[i]=&mat[i][0];
}
```



- By dynamic memory allocation, we do not require to declare two-dimensional array, it can be created dynamically using array of pointers.

```c
#include <stdio.h>

int main() {
    int numbers[] = {1, 2, 3, 4, 5};
    int *ptr = numbers; // Pointer to the first element of the array

    // Accessing array elements using the pointer
    printf("First element: %d\n", *ptr);    // Output: First element: 1
    printf("Second element: %d\n", *(ptr + 1));  // Output: Second element: 2

    // Using pointer arithmetic to traverse the array
    printf("Array elements: ");
    for (int i = 0; i < 5; ++i) {
        printf("%d ", *(ptr + i));
    }
    // Output: Array elements: 1 2 3 4 5

    return 0;
}

```