UNIT 5

Module – 5: (Pointers& File Handling)

- Pointers
- Introduction to Macros, Preprocessors
- Introduction to Dynamic memory allocation
- malloc(),calloc() and free() functions
- Notion to Linked lists
- Intoduction to file handling
- How to read,open,close files.
- Basic Program/s on file handling

Short Questions & Answers

Ques 1.What are Macros? Give examples.

Ans. Macros are the identifiers that represent statements or expressions. To associate meaningful identifiers with constants, keywords, and statements or expressions, **#define** directive is used.

If we define a macro i. e **SQUARE(x) x*x.** Here the macro determines the square of the given number.

Ques 2. What is Macro template?

Ans. Macro template is defining the macro at the beginning of a program using **'#'directive**. For example

#define CUBE (X) (X * X* X) .

Ques 3. What are the types of Macro?

Ans. (a) Object type Macro (b) Function types Macro.

Ques 4.What is undef directive?

Ans. This cause the specified identifier to be no longer defined as a macro name. For example:

.# undef PI

Ques 5.What is the difference between #include<stdio.h> and #include"stdio."h.

Ans. These are source file inclusion directives, the first way of inclusion searches the prespecified list of directories for the source file within < >.The second way the source file is first searched in the current working directory, if the search fails then prespecified list of directories are searched .If still the search fails,then compiler gives an error unable to include file.

Long Questions & Answers

Ques 6.What is dynamic memory allocation? What are malloc() and calloc() and free()functions?

Ans. Dynamic memory allocation deals with allocation & de-allocation of memoru space at the time of execution of a program**.** This technique is used where number of data nd information is not known beforehand and it may increase or decrease in future depending on certain situations. For Example

Let a company launches new model of a car and starts the booking. The number of customers interested in booking the car is not known apriori. So here structures, arrays can be allocated memory at run time rather than at compile time.

- The memory allocated by system to a program is divided into three portions (i) Memory for functions (ii) Memory for data (iii) Free memory called ***heap*** which may be required during execution of the program.
- Some ***library functions*** used in dynamic memory allocation have their ***prototype definition*** in header file **<*stdlib.h*>.**These functions are as follows:-
- **malloc()**:This stands for ***memory*** allocation This function allocates a block of memory of the size of its argument in bytes. It returns a void pointer to the first byte of the memory block

Synatx: void * malloc (size) ;

e.g: int *ptri;

ptri= (int*) malloc (n *sizeof(int)) ;

The function ***malloc allocates n*sizeof(int)*** bytes of memory and returns the value of pointer ptri.

Return value of malloc is the value of ptri which is the ***address of the first byte*** of the memory block.Let n=2 , then total bytes occupied are ***2 x 2=4 bytes.*** This is how an array can be allocated memory at run time. In case allocation is not possible due to lack of sufficient memory , ther return value will be NULL pointer .For example Code can be:

```
if (ptri= = NULL)
 printf ("Error in memory allocation") ;
exit(1) ;
```

Similarly for ***character array*:** char * ptrch;

ptrch= (char*) malloc (n *sizeof(char)) ;

In the same way we can create ***n structures dynamically*** as follows:

```
struct  book
{
   cahr name[15] ;
   int pages ;
 }
 struct  book*ptrb ;
 ptrb = (struct book*) malloc(n*sizeof(struct book)) ;
```

(ii) calloc( ): This stands for ***calculated*** allocation.This function is used to allocate an array of memory blocks and initializes each block member to 0. It returns a void pointer to the first byte of the first block of the memory. As compared to malloc function ***calloc has two arguments.***

Syntax: void* calloc (n , size_mem) ; // ***Here n is no.of blocks; size_m is size of each block***

(iii) realloc( ) : This function reallocates a memory block of size smaller or bigger than the size of the block earlier allocated by malloc().

Syntax: void* realloc (void *ptr , new_size) ;

```
e.g:  float *ptrf = (float *) malloc ( 5 * sizeof(float) ) ;  // allocates 5 x 4= 20 bytes of an array
      ptrf = (float *) realloc ( ptrf, 3 * sizeof(float) ) ;    // resized array of 3x 4=12 bytes.
```

(iv) free( ) : This function is used to de-alloacte thememory allocated by the functions malloc() , calloc() and realloc().The whole allocated memory block is released.

Syntax: void free (void *ptr) ;

Ques 7. What is File in C? What are different types of file used in file handling?

Ans. A **file** is a collection of bytes stored on a secondary storage device, which is generally a disk of some kind. There are two kinds of files that programmers deal with text files and binary files.

A **text file** can be a stream of characters that a computer can process sequentially. It is not only processed sequentially but only in forward direction.

For this reason a text file is usually opened for only one kind of operation (reading, writing, or appending) at any given time.

A **binary file** is no different to a text file. It is a collection of bytes. In C Programming Language a byte and a character are equivalent.

Ques 8. What is linked List? Explain.

Ans : The major application of dynamic memory allocation lies in creation of link lists.The objects in the link lists are linked together not like as arrays sequentially one after another, rather objects can be stored randomly in memory , but each element in the list keeps the address of the next element in the list , or address of both the neighbours, i.e : the previous as well as the next in the list. Each location is called a ***node of link list*** .Each node has two fields "data and address". If the node contains only one memory address, it is called a ***singly linked list.*** But if it contains ***address of just previous & next node*** then it is known as a ***doubly linked list***.

Each node of a link list is created using structures where data member of a structure node is pointer to that structure.

Creation of a Singly Link List:

```
Struct List
{  char name [20] ;
   int number;
   struct List *nextnode ;
} ; struct List *First ;          // First is a pointer of the type struct List.
```

Now the structure object pointed to by First must be allocated memory so as to store the data items. This is Implemented using malloc()as given below:

```
First = (struct List*) malloc (sizeof (struct List)) ;
typedef struct List Node ;
First = (Node*) malloc (sizeof(Node)) ;
```

The data items in this node may be initialized, as well as, a memory block may be created for the next node.

```
First -> name = "Dennis Ritchie";
First -> number = 80 ;
First -> Next = (Node* malloc) (sizeof (Node)) ;
```

Ques 9. Write statement in C language to open and close a file.

Ans : <u>**Opening a file:**</u>

The general format of the function used for opening a file is

```
FILE *fp;
fp=fopen("filename","mode");
```

The first statement declares the variable fp as a pointer to the data type FILE. As stated earlier, File is a structure that is defined in the I/O Library. The second statement opens the file named filename and assigns an identifier to the FILE type pointer fp. fopen() contain the file name and mode (the purpose of opening the file).

r is used to open the file for read only.
w is used to open the file for writing only.
a is used to open the file for appending data to it.

<u>**Closing a File**</u>

A file must be closed as soon as all operations on it have been completed. This would close the file associated with the file pointer. The input output library supports the function to close a file.

Syntax to close file

fclose(file pointer);

Example:

```
#include<stdio.h>
      #include <conio.h>
      void main(void)
      {
         FILE *myfile;
        char c;
        myfile = fopen("firstfile.txt", "r");


        if (myfile == NULL)

               printf("File doesn't exist\n");
        else   {
             do  {
                    c = getc(myfile);
                     putchar(c);  } while (c != EOF); }
               fclose(myfile);
           }
```

Ques 10.What are various File operations functions in C:

Ans : The various file operations are:

Function Name	**Operation**
fopen()	Creates a new file. Opens an existing file.
fclose	Closes a file which has been opened for use
getc()	Reads a character from a file
putc()	Writes a character to a file
fprintf()	Writes a set of data values to a file
fscanf()	Reads a set of data values from a file
getw()	Reads a integer from a file
putw()	Writes an integer to the file
fseek()	Sets the position to a desired point in the file

ftell()	Gives the current position in the file
rewind()	Sets the position to the beginning of the file

Ques 11. What types of errors may occur in file handling?

Ans. (a) Omission of File Pointer.

(b) Opening a File which does not exist.

(c) Reading a File which is in write mode.

(d) Writing into a file which is in read mode.

Ques 12. Write a C program to create a file called emp.rec and store information about a person, in terms of his name, age and salary.

Ans.

```
  #include <stdio.h>
 #include <conio.h>
void main()
{
            FILE *fptr;
            char name[20];
              int age;
            float salary;

             fptr = fopen ("emp.rec", "w"); /*open for writing*/
              if (fptr == NULL)
                {
                        printf ("File does not exists\n");
                      return ;
                }

   printf("Enter the name\n");
   scanf("%s", name);
   fprintf(fptr, "Name    = %s\n", name);

   printf("Enter the age\n");
   scanf("%d", &age);
   fprintf(fptr, "Age     = %d\n", age);

   printf ("Enter the salary\n");
   scanf("%f", &salary);
  fprintf( fptr , "Salary  = %.2f\n", salary) ;
  fclose(fptr);
}
```

```
/*-----------------------------------------------------------------------
Output
Enter the name
Prabhu
Enter the age
25
Enter the salary
25000
-------------------------------------
```

Please note that you have to open the file called emp.rec in the directory

[END of 5th UNIT]