Exp. No. 10

Aim:- Programs to learn Pointer operations.

Software Required:- Turbo C/C++

Theory:- A pointer is a variable whose value is the address of another variable, i.e. direct address of the memory location. Like any variable or constant, you must declare a pointer before using it to store any variable address. The general form of a pointer variable declaration is -

```
type *var-name;
```

Here, type is the pointer's base type; it must be a valid C data type & var-name is the name of the pointer variable. The asterisk * used to declare a pointer is the same asterisk used for multiplication. However, in the statement the asterisk is being used to designate a variable as a pointer. Take a look at some of the valid of the pointer declarations -

```
int *ip;    /* pointer to an integer */
double *dp; /* pointer to a double */
float *fp;  /* pointer to a float */
char *ch;   /* pointer to a character */
```

Instructer's Sign

Output of Code 1

Address of var variable : 28ff18
Address stored in ip variable : 28ff18
value of *ip variable : 20

Practical No.

PAGE NO. 62
DATE :

Code :-

```
1. #include<stdio.h>
   #include<conio.h>
   main()
      {
      int var = 20;
      int *ip;
      clrscr();
      ip = &var;
      printf("Address of var variable : %p\n", &var);
      printf("Address a stored in ip variable : %p\n",
                                        , ip);
      printf("value of *ip variable : %d\n", *ip);
      getch();
      }
```

Output of the code 2.

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

Practical No.

2.
```
#include <stdio.h>
#include <conio.h>
void main()
{ int var;
  int *ptr;
  int **pptr;
  clrscr();
  var = 3000;
  ptr = &var;
  pptr = &ptr;
  printf("Value of var = %d\n", var);
  printf("Value available at *ptr = %d\n", *ptr);
  printf("Value available at **pptr = %d\n",
                                **pptr);
  getch();
}
```

Output of Code 3

a = 2
b = 3
a = 2
b = 3

Practical No.

PAGE NO. 64
DATE :

3.
```
#include <stdio.h>
#include <conio.h>
void main()
{ int func(int a, int b);
  {
  printf("\n a = %d \n", a);
  printf("\n b = %d \n", b);
  return 0;
  }
void main()
  {
  int (*fptr) (int, int);
   fptr = func;
   func(2,3);
   fptr(2,3);
    getch();
  }
```

- **Output of Code 4**

```
Enter number of elements: 5
Enter elements of array:
4
8
6
2
1
Elements of array are:-
4 8 6 2 1
```

Practical No.

PAGE NO. 62
DATE :

4.

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
void main()
{
int n, i, *ptr, sum=0;
print("Enter number of elements: ");
scanf("%d", &n);
ptr =(int *)
malloc(n * sizeof(int));
if (ptr == NULL)
{
printf("Error! memory not allocated.");
return 0;
}
printf("Enter elements of array:\n");
for(i=0; i<n; ++i)
{
scanf("%d", ptr+i);
    sum += *(ptr+i);
}
printf("%d", ptr[i]);
    ptr[i];
}
free(ptr);
  (ptr);
getch();
}
```

Instructer's Sign

Viva Question

1. What do you mean by pointer?

Ans In C, A pointer is a variable that ~~store~~ stores the memory address of another variable. We use pointers to access the memory of the said variable & then manipulate their addresses in a program.

2. What is the role of * operator?

Ans The * operator in C has multiple roles, including (i) access the value stored at a pointer address. It is called the ~~def~~ "dereference".
(ii) The * operator performs multiplication on two operands.

3. What do you mean by the term pointer to pointer?

Ans Pointer to a pointer is a form of multiple indirection, or a chain of Pointer. Also known as a double pointer, is a variable that stores the memory address of another pointer variable.

4. How will passing the address or pointer in a function - call affect the parameters in the program?

Ans The called function can modify the value of the variable to which the pointer argument points. Pass-by-pointer means to pass a pointer argument in the calling function to the corresponding formal parameter of the called functi

Instructer's Sign

5. Is the array-name itself a pointer?

Ans Array names are not a pointer. Array & pointers are closely related & are often considered same by many people but this a common misconception. It is actually a label the refers to a contiguous block of memory where the element are stored.

Instructer's Sign

11

Aim:- Programs to understand File handling operations.

Software Required:- Turbo C / C++

Theory:- In programming, we may require some specific input data to be generated several numbers of times. Sometimes, it is not enough to only display the data on the console. The data to be displayed may be very large, and only a limited amount of data can be displayed on the console, and since the memory is volatile, it is impossible to recover the programmatically generated data again & again. However, if we need to do so, we may store it into the local file system which is volatile and can be accessed every time. Here, comes the need of the file handling in C.

File handling in C enables us to create, update, read and delete the files stored on the local file system through our C program. The following operations can be performed on a file.

- Creation of the new file
- Opening of the existing file
- Reading from the file.
- Writing to the file
- Deleting the file

Instructer's Sign

Practical No.

PAGE NO. 89.
DATE :

Output of the Code :

A text file program.txt is created in the C drive.

Enter num : 89 Console

program.txt

89

Code :-

```
1. #include <stdio.h>
#include <conio.h>
#include <stdlib.h>
  void main()
   {
   int num;
   FILE *fptr;
   clrscr();
   fptr = fopen ("C:\\program.txt", "w");
   if (fptr == NULL)
     {
     printf ("Error !");
     exit(1);
     }
   printf ("Enter num: ");
   scanf(" %d", &num);
   fprintf(fptr, "%d", num);
   fclose (fptr);
   getch();
  }
```

Output of the Code 2

Value of num has been read from file. "program.txt" present in C drive.

Value of n = 89

Practical No.

PAGE NO. 70.
DATE :

2.
```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
void main()
{
int num;
FILE *fptr;
clrscr();
if (fptr = fopen("C:\\program.txt","r")) == NULL)
  {
   printf("Error! opening file");
   exit(1);
  }
fscanf(fptr, "%d", &num);
printf("value of n = %d", num);
fclose(fptr);
getch();
}
```