9. POINTERS

EXERCISES

A] Answer the following questions.

1) What are pointers? Why they are important?

A pointer is a memory variable that stores a memory address. Pointer can have any name that is legal for other variable and it is declared in the same fashion like other variable but it is always denoted by '*' operator.

The importance of pointers is explained in exercise (2).

2) Explain features of pointers.

1) Pointers save the memory space.
2) Execution time with pointer is faster because data is manipulated with the address i.e. direct access to memory location.
3) The memory is accessed efficiently with the pointers. The pointer assigns the memory space and it releases. Dynamically memory is allocated.
4) Pointers are used with data structures. They are useful for representing two-dimensional and multi-dimensional arrays.

3) Explain pointer of any data type that requires four bytes.

A memory address of any variable is an unsigned long integer. It occupies 4 bytes. Hence, pointer of any data types that requires four bytes.

4) Explain use (*) indirection operator.

The (*) indirection operator is used to indicate pointer variable.

5) Explain effect of ++ and – operator with pointer of all data type.

The (++) increment operator when applied with pointer it indicates next memory location of its type. When (--) decremented it indicates previous memory location of its type.

6) What is an array of pointer? How it is declared?

Array of pointers is nothing but a collection of addresses. Here, we store address of variables for which we have to declare array as pointer. The following program explain this

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

void main()
{

int *arrp[3];
int arr[3]={5,10,15},k;

for (k=0;k<3;k++)
arrp[k]=arr+k;

clrscr();
printf ("\n\tAddress  Element\n");

for (k=0;k<3;k++)
{
        printf ("\t%u",arrp[k]);
        printf ("\t%7d \n",*(arrp[k]));
}

}
```

OUTPUT:

```
Address  Element
4060       5
4062       10
4064       15
```

Explanation: In the above program ****arrp[3]*** is declared as array of pointer. Using first ***for*** loop addresses of various elements of array ***'arr [ ]'*** are assigned to ***'*arrp[]'.*** The second ***for*** loop picks up addresses from ***'*arrp[]'*** and displays the value present at that locations. Here, each element of ***'*arrp[]'*** points to respective element of array ***'arr[]'***.

Element no.	Array of values	Element no.	Array of addresses
arr[0]	**5**	**arrp[0]**	**4060**
arr[1]	**10**	**arrp[1]**	**4062**
arr[2]	**15**	**arrp[2]**	**4064**

7) Explain relation between array and pointer.

The array name itself is a pointer. The array name points to first element of the array.

8) Why addition of two pointers is impossible?

The pointer holds address of another variables. Hence, addition of addresses is not possible.

9) Which are the possible arithmetic operations with pointers?

Arithmetic operations on pointer variables are also possible. Increment, decrement, prefix, & postfix operations can be performed with the pointers. The effect of these operations are shown in the below given table.

DATA TYPE	INITIAL ADDRESS	OPERATION		ADDRESS AFTER OPERATIONS		REQUIRED BYTES
int i=2	4046	++	--	4048	4044	2
char c='x'	4053	++	--	4054	4053	1
float f=2.2	4058	++	--	4062	4054	4
long l=2	4060	++	--	4064	4056	4

Table Pointer and Arithmetic Operation

From the above table we can observe that on increment of the pointer variable for integers the address is incremented by two i.e. ***4046*** is original address and on increment it's value will be ***4048*** because integers require two bytes.

Similarly, for characters, floating point numbers and long integers requires ***1, 4*** and ***4*** bytes respectively.

10) How one pointer points to another pointer?

Pointer is known as a variable containing address of another variable. The pointer variables also have an address. The pointer variable containing address of another pointer variables is called as pointer to pointer. This chain can be continued to any extent.

The below given program illustrates the concept of pointer to pointer.

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

void main()
{

int a=2,*p,**q;

p=&a;
q=&p;

clrscr();
printf  ("\n          Value of a= %d Address of a=%u",a,&a);
printf  ("\n Through    *p Value of a= %d Address of a=%u",*p,p);
printf  ("\n Through  **q Value of a= %d Address of a=%d",**q,*q);

}
```

OUTPUT:

```
                Value of  a= 2 Address of a=4056
Through   *p  Value of  a= 2 Address of a=4056
Through  **q   Value of  a= 2 Address of a=4056
```

Explanation: In the above program variable ***'p'*** is declared as pointer. The variable ***'q'*** is declared as pointer to another pointer. Hence, they are declared as **'*p'** and **'**q'** respectively. The address of variable ***'a'*** is assigned to pointer ***'p'***. The address of pointer ***'p'*** is assigned to ***'q'***. The variable ***'q'*** contains address of pointer variable. Hence, ***'q'*** is pointer to pointer. The program displays value and address of variable ***'a'*** using variable itself and pointers ***'p'*** and ***'q'***. The ***table 9.3*** illustrates the contents of different variables used in the program.

11) How will you recognize pointer to pointer? What the number of '*'s indicates.
The pointer of pointer is preceded by '**' in declaration.

12) How strings are stored in pointer variables? Is it essential to declare length?
The strings are stored in continuous memory locations. It is not essential to declare length.

13) What is base address? How it is accessed differently for one-dimensional and two-dimensional array.

14) Elaborate between the address stored in the pointer and the value at that address.

15) Why element counting of arrays always starts from '0'.
Because, the array name itself point to the address of first element. Hence, adding value greater than zero to pointer points the next element. Hence, 0 is added to access the first element.

16) Write a program to read and display a two-dimensional array of 5 by 2 numbers. Reduce base address of an array by one and start element counting from one.

B] Answer the following by selecting the appropriate option. (Answers in Bold)

1) Which of the following statement is true after execution of following program?

```
int a[5]={2,3},*c;
c=a;
(*c)--;
```

a) The value of a[0] will be 1;
b) The value of a[0] will be 2;
c) The value of a[1] will be 2;
d) None of the above.

2) The fastest way to exchange two rows in a two-dimensional array is

a) Exchange the addresses of each element in the two rows.
b) Exchange the elements of the two rows.
c) Store the addresses of the rows in an array of pointers and exchange the pointers.
d) None of the above

3) Which is the correct way to declare a pointer?

a) int *ptr; b) * int ptr; c) int ptr*; d) int_ptr x;

4) What will be the result of the following program?

```
void main()
{
int a=8,b=2,c,*p;
c=(a=a+b,b=a/b,a=a*b,b=a-b);
p=&c;
   clrscr();
   printf ("\n %d",++*p);
}
```

a) The value *p displayed will be 46.
b) The value *p displayed will be 48.
c) The value *p displayed will be 56.
d) The value *p displayed will be 40.

5) What will be the resulting string after execution of following program?

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

main()
{
    char *str1, *str2, *str3;
    str1="The Capital of India is ";
    str2="!!ihleD weN";
    str3="Bangalore";

    strncat(str1,strrev(str2),strlen(str3));
    clrscr();
    puts(str1);
}
```

a) The Capital of India is New Delhi
b) The Capital of India is New Delhi!!
c) The Capital of India is Bangalore
d) None of the above.

6) What will be the values of variables a and b after execution?

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

void main()
{
int a,*b=&a,**c=&b;
a=5;
**c=15;
*b= **c;
clrscr();
printf ("A=%d, B=%d",a,*b);
}
```

a) A=15, B=15
b) A=15,B=5
c) A=15, B=16
d) None of the above.

1) What will be the value of variable a1 and a2 after execution?

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

 main()
{
     int a1,a2,c=3,*pt;
    pt=&c;
    a1=3*(c+5);
    a2=3*(*pt+5);
}
```

a) A=24, B=24
b) A=12, B=24
c) A=12, B=24
d) None of the above.

8) What will be the value of x after execution of the following program?

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

void main()
{
int x,*p;
p=&x;
*p=2;
clrscr();
printf ("\nValue of x=%d",x);

}
```

a) x=2 b) x=0 c) x=65504 d) none of the above

C] Attempt the following programs.

1) Write a program to accept string using character pointer and display it.

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

 void main()
 {
    char *c;
    clrscr();
    printf ("\n Enter text : ");
    scanf ("%s",c);
    printf ("\n Entered text : %s",c);
 }
```

OUTPUT

Enter text : C_PLUS_PLUS

Entered text : C_PLUS_PLUS

Explanation In the above program ***the*** character pointer c is declared. Using scanf () string is read. The printf () statement displays the string.

2) Write a program to calculate square and cube of entered number using pointer of the variable containing entered number.

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

 void main()
 {
  int *n,a,s,c;
  clrscr();
  n=&a;
  printf ("\n Enter a number  : ");
  scanf ("%d",&a);
  s=pow(*n,2);
  c=pow(*n,3);
  printf ("\n Square : %d",s);
  printf ("\n Cube   : %d",c);
 }
```

OUTPUT

Enter a number : 3

Square : 9
Cube : 27

Explanation The *n is an integer pointer. It points to integer variable a. The entered integer entered is stored in variable a. using pow () function square and cube is calculated.

3) Write a program to display all the elements of an array-using pointer.

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

 void main()
 {
  int n[5],*p,j;
  clrscr();
  printf ("\n Enter a number  : ");
  for (j=0;j<5;j++)
  scanf ("%d",&n[j]);
  p=n;

  for (j=0;j<5;j++)
  printf (" %d ",*(p+j));
 }
```

OUTPUT
Enter a number : 3 4 2 5 4
3 4 2 5 4

Explanation An integer array and pointer are declared. The base address of array is stored in pointer n. By defereneeing the pointer, successive elements are obtained and displayed.