**Pointers:-**

Value = %d for signed, %u for unsigned, %f float, %ld long, %hd short, %lf double, %c character

I.Pointer assignment

1. int a=10;

int \*ptr;

ptr=&a;

a.print the value of a

b.print the address of a (&a)

c.print the value at ptr (\*ptr)

d.print the address stored at ptr (ptr)

e.print the address of ptr (&ptr)

2. char a=’A’;

char \*ptr;

ptr=&a;

a.print the value of a

b.print the address of a

c.print the value at ptr (\*ptr)

d.print the address stored at ptr (ptr)

e.print the address of ptr (&ptr)

3. Do the above code for float and double. (Use proper format specifiers)

II.Sizeofpointer(ptr) and \*ptr

4. int \*ptr1;

char \*ptr2;

double \*ptr3;

float \*ptr4;

printf("\n sizeof(ptr1)=%d", sizeof(ptr1));

printf("\n sizeof(ptr2)=%d", sizeof(ptr2));

printf("\n sizeof(ptr3)=%d", sizeof(ptr3));

printf("\n sizeof(ptr4)=%d", sizeof(ptr4));

printf("\n\n sizeof(\*ptr1)=%d", sizeof(\*ptr1));

printf("\n sizeof(\*ptr2)=%d", sizeof(\*ptr2));

printf("\n sizeof(\*ptr3)=%d", sizeof(\*ptr3));

printf("\n sizeof(\*ptr4)=%d", sizeof(\*ptr4));

III.Changing the pointer value

5. int a=10;

int \*ptr;

ptr=&a;

\*ptr = 20

a.print the value of a

b.print the value at ptr (\*ptr)

IV.pointer arithmetic

6. int a=10;

int \*ptr;

ptr=&a;   
printf(“\n%p”,ptr);  
ptr++; // ptr++ == ptr = ptr +1 printf(“\n%p”,ptr);

Do the above code for char, float, double data types and

Note down by how much the ptr increments.

7. int a=10;

int \*ptr;

ptr=&a;   
printf(“\n%p”,ptr);  
ptr--; // ptr-- == ptr = ptr -1 printf (“\n%p”,ptr);

Do the above code for char, float, double data types and

Note down by how much the ptr decrements.

8. int x=10; int y=20;   
int \*ptr1, int \*ptr2; ptr1=&x;ptr2=&y;

a.print the value of x and y

b.print the address of x and y (&x and &y)

c.print the value at ptr1 and ptr2 (\*ptr1 and \*ptr2)

d.print the address stored at ptr1 and ptr2 (ptr1 and ptr2)

\*ptr1 = y;

\*ptr2 = x;

Again,

a.print the value of x and y

b.print the address of x and y (&x and &y)

c.print the value at ptr1 and ptr2 (\*ptr1 and \*ptr2)

d.print the address stored at ptr1 and ptr2 (ptr1 and ptr2)

9. int x=10; int y=20;   
int \*ptr1, int \*ptr2; ptr1=&x; ptr2=&y;

a. print the value of x and y

b. print the value at ptr1 and ptr2 (\*ptr1 and \*ptr2)

c. print the address stored at ptr1 and ptr2 (ptr1 and ptr2) ptr1=ptr2;

Again,

a. print the value of x and y

b. print the value at ptr1 and ptr2 (\*ptr1 and \*ptr2)

c. print the address stored at ptr1 and ptr2 (ptr1 and ptr2)

V.\*ptr++ ,\*(++ptr), ++\*ptr and (\*ptr)++

10.intnum=10;

int \*ptr;

ptr=&num;

printf("\n\*ptr=%d",\*ptr);

printf("\n++\*ptr=%d",++\*ptr);

Try for (\*ptr)++

11. intnum=10;

int \*ptr;

ptr=&num;

printf("\n ptr=%p",ptr);

printf("\n\*ptr=%d ",\*ptr);

printf("\n\*ptr++=%d",\*ptr++);

printf("\n ptr=%p",ptr);

Try for \*(++ptr)

VI.Const Pointer ans pointer to const

12. intnum=10;   
int \*constptr = &num;   
printf("\n Changing the address ptr=%p", ptr++); is this valid ??

printf("\n changing the value \*ptr=%d ",++\*ptr);

13. constintnum=10;

constint \*ptr;

ptr=&num;

printf("\n Changing the address ptr=%p", ptr++);

printf("\n changing the value \*ptr=%d ",++\*ptr); is this valid ??

14. constintnum = 10;

constint \*constptr=&num;

printf("\n changing the value \*ptr=%d ",++\*ptr); is this valid ?

printf("\n changing address ptr=%p", ptr++); is this valid ? printf("\n ptr=%p \*ptr=%d", ptr,\*ptr);

VII.Void pointer

15.inti;

char c;

void \*data;

i = 6;

c = 'a';

data = &i;

printf("data points to the integer value %d\n", \*(int\*)data);

data = &c;

printf("data now points to the character %c\n", \*(char\*) data);

VIII.Double pointer

16.   
intnum=10;

int \*ptr;

int \*\*ptr1;

ptr=&num;

ptr1=&ptr;

printf("\n value of num=%d", num);

printf("\n address of num=%p", &num);

printf("\n\n value at \*ptr=%d",\*ptr);

printf("\n address at ptr=%p", ptr);

printf("\n address of ptr=%p", &ptr);

printf("\n\n value at \*\*ptr1=%d",\*\*ptr1);

printf("\n address at ptr1=%p",ptr1);

printf("\n address of ptr1=%p",&ptr1);

17. char \*ptr = “hello”;

printf(“%d”,++\*(ptr));

printf(“%d”,(\*ptr)++);

**Structure:-**

1.Declare a structure to store data for student information. The structure contains roll number, name and percentage of students. Declare structure variable. Write a program to accept information of student from user and print the same.

2.Declare a structure to store data for student information. The structure contains roll number, name and percentage of students. Declare array of structure for storing 5 students data. Write a program to accept information of students from user and print the same.

3.a. Declare a structure to store data for student information. The structure contains roll number, name and percentage of students. Declare structure variable. Declare the structure pointer. (Pointer pointing to structure variable). Write a program to accept information of student from user and print the same.(Using pointer - > operator)

3. b. Declare a structure to store data for student information. The structure contains roll number, name and percentage of students. Declare array of structure for storing 5 students data. Declare the structure pointer (pointer pointing to structure array). Write a program to accept information of students from user and print the same. (Using pointer - > operator )

4.Repeat question no 1 . Write a function to accept information of student from user. Write another function to print the information. (Pass entire Structure to Function(call by address )) - > operator

5.Do the all above questions using typedef method.