Understand these questions and explore what happening.

1)

```
#include <stdio.h>
#include <string.h>
int main()
{
    char str1[20] = "Hello";
    char str2[20] = "World";
    char *p1 = str1, *p2 = str2;
    while (*p1)
    {
        p1++;
    }
    while (*p2)
    {
        *p1 = *p2;
        p1++;
        p2++;
    }
    *p1 = '\0';
    printf("%s", str1);
    return 0;
}
```

2)

```
#include<stdio.h>

int main()
{
  int a;
  int* p = &a;
  printf("%zu", sizeof( *(char*)p ));
}
```

3)

```
#include<stdio.h>

fun(int a)
{
  char *arr[] = {"0000",
"0001","0010","0011","0100","0101","0110","0111","1000","1001","1010","1011","1
100","1101","1110","1111"};
  unsigned char* p = (unsigned char*) &a ;
  p+=3;
  int i;
  for(i = 0; i < sizeof a; i++)
  {
    int d = (*p)>>4;
    printf("%s", arr[d]);
    d = (*p) & 0xf;
    printf("%s ", arr[d]);
    p--;
  }
}

int main()
```

```
    {
    int a;
    scanf("%d", &a);
    fun(a);
    }
```

4)

```
    #include<stdio.h>
    void main ()
    {
        int val1, val2;
        int *ptr =  0;
        if (ptr == 0)
        {
            ptr = &val1;
            val1 = 5;
        }
        if (ptr ==  0)
        {
            ptr = &val2;
            val2 = 10;
        }
        printf ("value of *ptr: %d ", *ptr);
    }
```

5)

```
        char ch = 10;
        void* ptr = &ch;
        printf("%d, %d", *(char*)ptr, ++(*(char*)ptr));
```

6)

```
    int a=2,*f1,*f2;
    f1=f2=&a;
    *f2+=*f2+=a+=2.5;
    printf("\n%d %d %d",a,*f1,*f2);
```

7)

```
    int i = 257;
    int *iPtr = &i;
    printf("%d %d", *((char*)iPtr), *((char*)iPtr+1) );
```

8)

```
    #include <stdio.h>
    void print(char *p);
    int main()
    {
    char s[] = "T.C.S", *A;
    print(s);
    return 0;
    }
    void print(char *p)
    {
    while (*p != '\n')
    {
    if (*p != '.')
    printf ("%c", *p);
    p++;
    }
    }
```

9)

```
#include<stdio.h>
void main()
{
int i,count=0;
char *p1="abcdefghij";
char *p2="alcmenfoip";

for(i=0;i<=strlen(p1);i++) {
          if(*p1++ == *p2++)
   count+=5;
  else
   count-=3;
}
printf("count=%d\n",count);
}
```

10)

```
void main()
{
char *p1="name";
char *p2;
p2=(char*)malloc(20);
memset (p2, 0, 20);
while(*p2++ = *p1++);
printf("%s\n",p2);
}
```

11)

```
int a[10];
printf("%d",*a+1-*a+1);
```

12)

```
int arr[3]={10,20,30};
int x=0;
x=++arr[++x]+ ++x+arr[--x];
printf("%d ",x);
```

13)

```
int a[]={10,20,30,40};
int i=3,x;
x=1*a[--i]+2*a[--i]+3*a[--i];
printf("%d",x);
```

14)

```
static char *s[] = {"black", "white", "yellow", "violet"};
char **ptr[ ] = {s+3, s+2, s+1, s}, ***p;
p = ptr;
***++p;
printf("%s",++*--*++p + 2);
```

15)

```
static int a[3][3]={1,2,3,4,5,6,7,8,9};
int i,j;
static int *p[]={a,a+1,a+2};
for(i=0;i<3;i++)
{
for(j=0;j<3;j++)
printf("%d\t%d\t%d\t%d\n",*(*(p+i)+j),*(*(j+p)+i),*(*(i+p)+j),*(*(p+j)+i));
}
```

16)

```
void main ()
{
int x [3][4] = {{1,6,9,12},{11,17,3,2},{20,23,4,5}};
int *n = &x;
}
```

(i) *(*(x + 1) + 1) = ?
(ii) *(*x + 1) + 3 = ?
(iii) *(n + 3) + 1 = ?
(iv) ++(*n++) + *n = ?

17)

```
char s[]={'a', 'b', 'c', '\n', 'c', '\0'};
char *p, *str, *str1;
p= &s[3];
str= p;
str1= s;
printf("%d", +++p+++str1-32);
```

18)

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
typedef struct
{
char *name;
} Addr;
int main()
{
Addr *s;
char comm[10];
char *str = "Hello";
s = (Addr *)malloc(sizeof(Addr));
printf("Enter a name");
fgets(comm, 10,stdin);
s->name = (char*)malloc(sizeof(char[strlen(comm)]));
strcpy(s->name, comm);
strcat(str,s->name);
printf("%s", str);
}
//for the input : india
```

19)

```
char *pl= "word";
char *p2;
p2=(char*)malloc(10);

memset(p2, 'A', 10);
while(*p2++ = *pl++)
printf("%s ", p2);
```

20)

```
#include <stdio.h>

struct node {
    int data;
    struct node *next;
```

```
};
void node_function(struct node **n) {
    (*n)->data = 2;
    (*n)->next->data = 3;
}
int main() {
    struct node n1 = {3, NULL};
    struct node n2 = {2, NULL};
    n1.next = &n2;
    struct node *ptr = &n1;
    node_function(&ptr);
    printf("%d %d", n1.data, n2.data);
    return 0;
}
```

21) What will be the output of the following code snippet for the list 1->2->3->4->5->6?

```
void solve(struct node* start)
{
if(start == NULL)
return;
printf("%d ", start->data);
if(start->next != NULL )
solve(start->next->next);
printf("%d ", start->data);
}
```

22)

```
int a;
char *x;
x = (char *) &a;
a = 512;
x[0] = 1;
x[1] = 2;
printf("%d",a);
```

23)

```
void function(char **ptr)
{
char *ptr1;
ptr1 = (ptr += sizeof(int))[-3];
printf("%s\n", ptr1);
}
int main()
{
  char *arr[] = { "cat", "bat", "rat", "dog", "egg", "fly" };
function(arr);
return 0;
}
```

24)

```
int *a;
a=(int*)malloc(sizeof(int));
     *a=100;
     printf("%d\n",*a);
    free(a);
```

```
        *a=200;
        printf("%d\n",*a);
```

25)

```
    struct p
        {
            int x[2];
        };
        struct q
        {
            int *x;
        };
        int main()
        {
            struct p p1 = {1, 2};
            struct q *ptr1;
            ptr1->x = (struct q*)&p1.x;
            printf("%d\n", ptr1->x[1]);
        }
```

26)

C program to change the value of constant integer using pointers.

27)

Write a program in C to find the factorial of a given number using pointers

28)

Write a program in C to demonstrate the use of pointers to structures.

29)

Write a program in C to print all the alphabets using a pointer.

30)

C program to multiply two matrix using pointers.