

C APTITUDE

```
1)main()
{
    int const *p = 10;
    printf("%d", ++(*p));
}
```

```
2)main()
{
    int i;
    char *a = "man";
    for(i=0;i<3;i++)
    printf("%c%c%c%c",i[a],a[i],*(a+i),*(i+a));
}
```

```
3) main()
{
    extern int i = 10;
    printf("%d",i);
}
```

```
4) main()
{
    float f= 1.1;
    double d = 1.1;
    if ( f == d)
    printf("I LOVE U");
    else
    printf("I HATE U");
}
```

```
5) Struct {
    int a:1;
}b={1};
int main() {
    printf("%d",b.a);
    Return 0;
}
```

```
6) fun(a,b){
    if(a>b)
    return(-1);
    else if(a<b)
    return(+1);
}
void main(){
    int a=12,b=10;
    if(a>b)
    fun(a,b);
    else if(a==b)
    return(0);}
```

```
7) void fun(int a,int b)
{
    While(1){
        c=a%b;
        If(c==0)
        return b;
        a=b;
        b=c; }
}
```

What does this above code does?

```
8) int main()
{
    int a[5]={1,2,3,4,5};
    int *ptr=(int*)(&a+1);
    printf("%d %d",*(a+1),*(ptr-1));
}
```

```
9) void main()
{
    Printf("\n%%dog\");
}
```

```
10) void main(){
    Char const * p="Compiler", *q="assembler";
    q="linker";
    Printf("p=%s q=%s",p,q);
}
```

```
11)void main() {
    Int *p=new int;
    *p=13;
    While((*p)--);
    Cout<<*p;
    Free(p);
}
```

```
12)int main(){
    int arr[3]={2,3,4};
    char *p;
    p=(char*)arr;
    p=(char*)((int*)(p));
    printf("%d",*p);
    p=(char*)(p+1);
    printf("%d\n",*p);
    return 0;
}
```

```
13) int main()
{
    char *str;
    str="%s\n";
    printf(str,"K\n");}
```

```

14) main(){
int c[]={2.8,3.4,4,6.7,5};
int j,*p=c,*q=c;
for(j=0;j<5;j++) {
printf(" %d ",*c);
++q; }
for(j=0;j<5;j++){
printf(" %d ",*p);
++p; }
}
15)main()
{
char s[]={'a','b','c','\n','c','\0'};
char *p,*str,*str1;
p=&s[3];
str=p;
str1=s;
printf("%d",++*p + ++*str1-32);
}
16)main()
{
int a[2][2][2] = { {10,2,3,4}, {5,6,7,8} };
int *p,*q;
p=&a[2][2][2];
*q=***a;
printf("%d---%d",*p,*q);
}
17) main()
{
struct xx
{
int x=3;
char name[]="hello";
};
struct xx *s;
printf("%d",s->x);
printf("%s",s->name);
}

18) main()
{
struct xx
{
int x;
struct yy
{
char s;
struct xx *p;
};
struct yy *q;
};
}

```

```

19) enum colors {BLACK,BLUE,GREEN}
main()
{
printf("%d..%d..%d",BLACK,BLUE,GREEN);
return(1);}

20)main()
{
char *p;
p="Hello";
printf("%c\n",&*p);
}

21) main()
{
int i=1;
while (i<=5)
{
printf("%d",i);
if (i>2)
goto here;
i++;
}
}
fun()
{
here:
printf("PP");
}

22) main()
{
static char names[5][20]={"pascal","ada","cobol",
"fortran","perl"};

int i;
char *t;
t=names[3];
names[3]=names[4];
names[4]=t;
for (i=0;i<=4;i++)
printf("%s",names[i]);
}

23)main( )
{
static int a[] = {0,1,2,3,4};
int *p[] = {a,a+1,a+2,a+3,a+4};
int **ptr = p;
ptr++;
printf("\n %d %d %d", ptr-p, *ptr-a, **ptr);
*ptr++;
printf("\n %d %d %d", ptr-p, *ptr-a, **ptr);
**ptr++;
printf("\n %d %d %d", ptr-p, *ptr-a, **ptr);
***ptr;
printf("\n %d %d %d", ptr-p, *ptr-a, **ptr);
}

```

```

24) main( )
{
void *vp;
char ch = 'g', *cp = "goofy";
int j = 20;
vp = &ch;
printf("%c", *(char *)vp);
vp = &j;
printf("%d",*(int *)vp);
vp = cp;
printf("%s", (char *)vp + 3);
}
25) main ( )
{
static char *s[] = {"black", "white", "yellow",
"violet"};
char **ptr[] = {s+3, s+2, s+1, s}, ***p;
p = ptr;
**++p;
printf("%s",*--*++p + 3);
}
26) int i,j;
for(i=0;i<=10;i++)
{
j+=5;
assert(i<5);
}
27) main()
{
main();
}
28) main()
{
char name[10],s[12];
scanf("%s",s);
}
How scanf will execute?
29) main()
{
char *str1="abcd";
char str2[]="abcd";
printf("%d %d
%d",sizeof(str1),sizeof(str2),sizeof("abcd"));
}
30) main()
{
int k=1;
printf("%d==1 is %s" ,k,k==1?"TRUE":"FALSE");
}

```