

3.Q Post increment

Sol:

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

```
int main()
{
    int a,b;
    a=10;
    b=a++;
    printf("\n a=%d",a);
    printf("\n b=%d",b);
}
```

Output

a=11

b=10

2.Q Pre increment

Sol:

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

```
int main()
{
    int a,b;
    a=10;
    b=++a;
    printf("\n a=%d",a);
    printf("\n b=%d",b);
}
```

Output

a=11

b=11

4.==,!= diff:

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

```
int main()
{
    int n;
    printf("\n plz enter 9: ");
    scanf("%d",&n);
    if(n==9) // comparison
        printf("\n Thank you ");
    if(n=9) // assignment
        printf("\n Sorry ");
}
```

Output

plz enter 9:

Thank you

Sorry

```
5./* conditional operators demo-1*/
```

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

```
int main()
{
    int n;
    printf("\n enter a no: ");
    scanf("%d",&n);
    printf((n>0)? "positive ":(n<0)? "negative ":"\nzero");
    return 0;
}
```

Output:

-22

enter a no: negative

22

enter a no: positive

6./* conditional operators demo-3*/

#include <stdio.h>

int main()

{

int a,b,c;

printf("\n enter a,b,c values: ");

scanf("%d%d%d",&a,&b,&c);

printf(a==b && b==c ?"all are equal " :a>b && a>c ? "a is big ":b>c ? "b is big " : "c is big");

return 0;

}

Output:

11

12

10

enter a,b,c values: b is big

7./* implicit type casting*/

#include <stdio.h>

int main()

{

int a=2.3;

float b=10;

char c=65;

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

printf("\n b= %d",b);

printf("\n c= %d",c);

}

Output

a=2

b=10.000000

c=A

```
8./* ASCII demo1*/
```

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

```
int main()
```

```
{
```

```
    int var=65;
```

```
    char ch='a';
```

```
    printf("\n var= %d",var);
```

```
    printf("\n var= %c",var);
```

```
    printf("\n ch= %c",ch);
```

```
    printf("\n ch= %d",ch);
```

```
}
```

Output:

```
var= 65
```

```
var= A
```

```
ch= a
```

```
ch= 97
```

```
9./* ASCII demo2 0 to 256 values*/
```

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

```
int main()
```

```
{
```

```
    for(int i=48;i<=122;i++)
```

```
    {
```

```
        printf("\n%d\t%c",i,i);
```

```
    }
```

```
    return 0;
```

```
}
```

Output:

```
48    0
```

```
49    1
```

```
50    2
```

```
51    3
```

```
52    4
```

```
53    5
```

```
54    6
```

55	7
56	8
57	9
58	:
59	;
60	<
61	=
62	>
63	?
64	@
65	A
66	B
67	C
68	D
69	E
70	F
71	G
72	H
73	I
74	J
75	K
76	L
77	M
78	N
79	O
80	P
81	Q
82	R
83	S
84	T
85	U
86	V
87	W
88	X
89	Y
90	Z

91	[
92	\
93]
94	^
95	_
96	`
97	a
98	b
99	c
100	d
101	e
102	f
103	g
104	h
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
117	u
118	v
119	w
120	x
121	y
122	z

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

```
int main()
{
    int r,s1,s2,s3,t;
    char n[20];
    float a;
    printf("\nenter the number:");
    scanf("%d",&r);
    printf("\nenter the name:");
    scanf("%s",&n);
    printf("\nenter the 3marks:");
    scanf("%d%d%d",&s1,&s2,&s3);
    t=s1+s2+s3;
    a=(float)t/3;
    printf("\ntotal:%d,average:%f",t,a);
    return 0;
}
```

Output:

56

bharadwaj

10

9

6

enter the rnumber:

enter the name:

enter the 3marks:

total:25,average:8.333333