Date ______
Page ______

→ Predict the following output

1.
```
#include <stdio.h>
void fun(int x)
{
    x = 30;
}
int main()
{
int y=20;
fun(y);
printf("%d", y);
return 0;
}
```

→ output: 20
Parameter are passed by value, so it does not change the value of y through fun^c.

2.
```
#include <stdio.h>
void fun(int *ptr)
{
    *ptr = 30;
}
int main()
{
 int y = 20;
 fun(&y);
 printf("%d", y);
 return 0;
}
```

→ output: 30
call by reference.
*ptr = 30 ~~changes~~ ~~the~~ is value of y
&y changes the value y to *ptr

Scanned by CamScanner

3.
```
int main( )
{
int *ptr;
int x;              x = 0
ptr = &x;
*ptr = 0;
printf ("x = %d\n", x);        = 0
printf("*ptr = %d\n", *ptr);   = 0
*ptr += 5;
printf ("x = %d\n", x);        5
printf ("~~*~~ *ptr = %d\n", *ptr);   5
(*ptr)++;
printf("x = %d\n", x);         6
printf("*ptr = %d\n", *ptr);   6
return 0;
}
```

→
```
output // Explanation
 x = 0  // value of x
*ptr = 0 // value at x
 x = 5  // value of x   i.e *ptr += 5; = 0+5 = 5
*ptr = 5 // value at x
 x = 6  // value of x  i.e after *ptr increments(6)
*ptr = 6 // value at x
```

4.
```
#include <stdio.h>
int main ( )
{
char s1[7] = "1234", *p;
p = s1 + 2;
*p = '0';
printf ("%s", s1);
}
```

```
5. #include<stdio.h>
void f(int *p, int *q)  // P points i, q points j
{
  P=q;                  // P also points to j
*p = 2;                 // value change to 2 at y.
}
int i=0, j=1; //initializia
int main()
{
f(&i, &j); // func call
printf("%d %d \n", i, j); //prints i=0, j=2
getchar();
return 0;
}
output: 02
```

Explanation

```
6. #include<stdio.h>
int f(int x, int *py, int **ppz)
{
 int y, z;
**ppz += 1;
 z = **ppz; = 5
*py += 2;
 y = *py; = 7
 x += 3; = 7
 return x+y+z;
}
void main()
{
int c, *b, **a;
 c = 4;
 b=&c;
 a = &b;
 printf("%d", f(c,b,a));
   return 0;
}
```

→ Output: 19
Explanation:
It changes the value of c to 5 i.e; $z = **p2$;
$y = *py = 7$, $x += 3 = 7$, $x = 7$, $y = 7$, $z = 5$
then it returns $x + y + z$;
$7 + 7 + 5$
$= 19$

7.
```
#include<stdio.h>
int main ( )
{
int arr[] = {1, 2, 3, 4, 5};
int *p = arr;
++*p;
p+= 2;
printf("%d", *p);
return 0;
```

$*p = arr = 1$
$++(*p) = 1$ (with 1 above p)
$p += 2 = 3$

→ O/P: 3
→ Explanation:
$*p = arr$ → points to 1st element i.e; 1
$++(*p)$ → points to 1st element & then increment the by value i.e; 2
$p += 2$ → points to the base address of 3rd element.
$*p$ holds the value of 3rd element.

8.
```
#include<stdio.h>
int main( )
{
char c[] = "GATE2011";
char *p = c;
printf("%s", p+p[3]-p[1]);
}
```

G	A	T	E	2	0	1	1
1000	1004	1008	1012	1016	1020	1024	1028

$= 1000 + 4 * 4$
$= 1016$

→ O/P : 2011

Explanation:

```
0 1 2 3 4 5 6 7
G A T E 2 0 1 1
```

69-65 = 4

P[3] is E - P[1] = A = 4 { ascii value of E & A

[illegible] P + 4 = 2011

9\.

```
int main()
{
char arr[] = "worksheet";
printf(" %s", P + P[1] - P[3]);
return 0;
}
```

W	o	r	k	s	t	r	e	e	t
1000	1004	1008	1012	1016	1020	1024	1028	1032	1036

P + P[1] - P[3]
= P + 4
= 1000 + 4 * 4
= 1016

O/P : "street

Explanation

```
0 1 2 3 4 5 6 7 8 9
w o r k s t r e e t
```

It prints from base address of 4th element

10\.

```
#include<stdio.h>
int fun(char *str1)
{
char *str2 = str1;
while (* ++str1);
return (str1-str2);
}
int main()
{
char *str = "workstreet";
printf("%d", fun(str));
return 0;
}
```

output : 10

%d prints the integer value.