Assignment

Sapna
PAGE NO.
DATE / /

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

o/p: 20

The parameter is passed by value. The value of y is not modified using the function fun(). The value changes only inside the function by creating separate variable. The original value of y remains same.

---

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

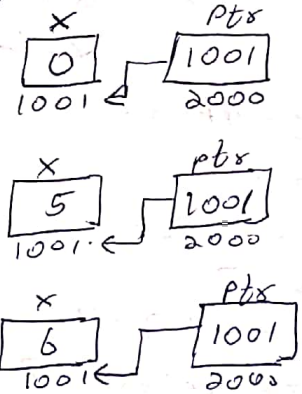
o/p: 30

The parameter is passed by reference.
The function modifies the value at the address ptr.
In the statement *ptr = 30, the value at address ptr is changed to 30.

3) int main()
{
int *ptr;
int x;
ptr = &x; // ptr points to x
*ptr = 0; // value at ptr is 0 or x is 0
printf("x = %d\n", x); // value of x is 0
printf("*ptr = %d\n", *ptr); // value at ptr is 0
*ptr += 5; // increment value at ptr by 5
printf("x = %d\n", x); // value of x is 5
printf("*ptr = %d\n", *ptr); // value at ptr is 5
(*ptr)++; // increment value at ptr by 1
printf("x = %d\n", x); // value of x is 6
printf("*ptr = %d\n", *ptr); // value at ptr is 6
return 0;
}

o/p: Warning for not using #include <stdio.h>

x = 0
*ptr = 0
x = 5
*ptr = 5
x = 6
*ptr = 6



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

o/p: 1204

s1 points to the first character of array.
p holds the address of 3rd character
The value at p is changed to 0
It prints 1204

Sapna
PAGE NO.
DATE / /

5) #include <stdio.h>
void f(int *p, int *q)
{
p = q;
*p = 2;
}
int i=0, j=1;
int main()
{
f(&i, &j);
printf("%d %d \n", i, j);
getchar();
return 0;
}

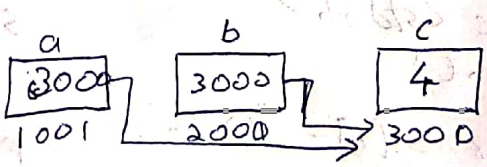
p points to the address of i
q points to the address of j
when, p = q; p also points to j.
*p = 2; The value of i is changed to 2
It prints i=0
j=2

O/P: 0 2

---

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

c is passed by value
a and b are passed by reference
x = 4;
// c = 4+1 = 5
// z = 5
// c = 5+2 = 7
// y = 7
// x = 4+3 = 7
It returns 7+7+5 = 19



O/P: 19

Sapna
PAGE NO.
DATE / /

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

O/P: 3

p points to the first element of array.
++*p is evaluated as ++(*p)
It increment the value of first element (1 as 2).

In p += 2;
The value of p is changed to address of third element.
Then it prints the value at third address of array.

---

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

O/P: 2011

p holds the base address of array c.
p[3] = 'E'
p[1] = 'A'
p[3] - p[1] = 69 - 65 = 4
(ASCII values)
p+4 is base address of string "2011"

---

9)
int main()
{
char arr[] = "Workstreet";
printf("%s", arr+4);
return 0;
}

O/P: street

arr+4 is the base address of string street.

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;
}
```

o/p: 10

The fun counts the number of characters in the string.

The pointer str2 is initialized as str1.

while (*++str1); increments str1 till '\0'

str1 is incremented by 10.

str1 - str2 = 10.