Branch : CSE & IT **Batch:Hinglish**

WEEKLY TEST – 05

Subject : Programming in C

Topic : Arrays and Pointers

Maximum Marks 20

Q.1 to 6 Carry ONE Mark Each

[MCQ]

1. Consider the following statements:

```
P: int a[2]={5,10};
   printf("%d", 1[a]);
Q: int 2[a]={5,10};
    printf("%d", 1[a]);
```

Which of the following statement(s) is/are INCORRECT?

(a) Both P and Q

(b) P only

(c) Q only

(d) Neither P nor Q

[MSQ]

2. Which of the following declaration(s) is/are NOT allowed?

(a) int b[][]={1,2,3,4};

(b) int b[][2][2]={1};

(c) int b[]={1};

(d) int b[][3];

[MCQ]

3. Consider the following program:

```
#include <stdio.h>

int main()

{

   int a[3][4]={1,2,3,4,5,6,7,8,9,10,11,12};

   printf("%d",*(*(a+**a+1)+2)+3);

   return 0;

}
```

The output is-

(a) 11 (b) 12

(c) 14 (d) Compilation Error

[NAT]

4. Consider the following program:

```
#include <stdio.h>

int main()

{

   int a[]={1,3,5,7,9};

   int i, count=0;

   int *b=a+4;

   for(i=0;i<3;i++)

   count=count+(*b---i);

   return 0;

}
```

The final value of count is _______.

[MCQ]

5. Consider the following program:

```
#include <stdio.h>

int main()

{

   int a[]={1, 2, 3, 4, 5};

   int *ptr=a;

    ptr+=sizeof(2*a[0]);

    printf("%d",*(ptr-2));

    return 0;
}
```

(Assume, integer size is 4 bytes)

The output is-

(a) 4 (b) 3
(c) Garbage value (d) Compilation Error

[MCQ]

6. Consider the following codes:

```
P: int *p=NULL;
   printf("%d", *p);
Q: int *p;
   *p=10;
   printf("%d", *p);
```

Which of the following is CORRECT?

(a) Neither P nor Q is valid.
(b) Only P is valid.
(c) Only Q is valid.
(d) Both P and Q are valid.

Q.7 to 13 Carry TWO Mark Each

[MCQ]

7. Consider the following program: 2 Marks

```
#include <stdio.h>

#include <stdlib.h>

int main() {

int i;

int *p=(int*)malloc(3*sizeof(int));

for(i=0;i<3;i++)

*(p+i)=3-i;

int *q=p;

printf("%u\t",*++p);

printf("%u\t", ++*p);

printf("%u\t", p-q);

return 0;

}
```

The output is-

(a) 1 2 3 (b) 3 2 1
(c) 2 2 1 (d) 2 3 1

[MCQ]

8. Consider the following program:

```
#include <stdio.h>
#include <stdlib.h>
int main() {
void *p, *q;
int k=97;
char b='C';
p=&k;
q=&b;
printf("%d", *(char*)p-*(char *)q);
return 0;
}
```

The output is-

(a) Garbage value
(b) Compilation error
(c) 30
(d) No output

[MCQ]

9. Consider the following program:

```
#include <stdio.h>

int func(int *p, int n){

int sum=*(p+4);

for(int i=1;i<n-2;i++){

sum=sum+*(p+i)+*p++;

}
return sum;
}
int main()
{
    int a[]={7, 1, 3, 5, 2};
    int (*ptr)(int *, int)=func;
    printf("%d",(*ptr)(a, 5));
    return 0;
}
```

The output is-

(a) 16
(b) 23
(c) Garbage value
(d) Compilation Error

[MCQ]

10. Consider the following functions:

```
P:   int * f1(){
         int x=10;
         return &x;
     }
Q:   int *f2(){
         static int x=10;
         return &x;
     }
R:   int *f3(){
         int x=10;
         int *p=&x;
         return *p;
     }
```

Which of the given functions is/are VALID?

(a) P (b) Q
(c) R (d) P and R

[NAT]

11. Consider the following program:

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

The output is ______.

[MCQ]

12. Consider the following program:

```
#include <stdio.h>
int f(int (*ptr)[3], int n)
{
    if(n<=1) return 0;
    return f(ptr+1, n-1)+**ptr;
}
int main()
{
    int a[][3]={1,2,3,4,5,6,7,8,9};
    printf("%d",f(a,3));
    return 0;
}
```

The output is-

(a) 5 (b) 11
(c) Garbage value (d) Compilation Error

[NAT]

13. Consider the following program:

```
#include <stdio.h>
int main()
{
```

```
int a[2][2][3]={0,1,2,3,4,5,6,7,8,9,10,11};
printf("%d", *(*(*a+1)+1)+1);
printf("%d", a[1][1][1]);
printf("%d", ***(a+1));
printf("%d", **(*a+1));
printf("%d", *(**a+1));
    return 0;
}
```

The sum of printed values is ___________

Answer Key

1. (c)
2. (a, d)
3. (c)
4. (18)
5. (b)
6. (a)
7. (d)
8. (c)
9. (a)
10. (b)
11. (8)
12. (a)
13. (25)

Hints and Solutions

1. (c)

P: int a[2]={5,10};//It is valid declaration.
printf("%d", 1[a]);//1[a] is equivalent to a[1].
Hence, P is correct.

Q: int 2[a]={5,10};//It is invalid declaration.
printf("%d", 1[a]);
Hence, Q is incorrect.

2. (a, d)

int b[][]={1,2,3,4};//Not allowed

int b[][2][2]={1};//allowed

int b[]={1};//allowed

int b[][3];//Not allowed

3. (c)

a denotes the base address of the 0th 1D array.
**a denotes the 0th element of the 0th 1D array.
((a+**a+1)+2)+3=*(*(a+1+1)+2)+3=*(*(a+2)+2)+3
a+2 points to the 2nd 1D array. *(a+2) points to the 0th element of the 2nd 1D array.
(*(a+2)+2) points to the 2nd element of the 2nd 1D array.

((a+2)+2) is the 2nd element of the 2nd 1D array. So, *(*(a+2)+2)=11

((a+2)+2)+3=11+3=14.
Output: 14

4. (18)

b points to the 4th element of a i.e 9.
For i=0:
count=count+(*b---i);//post decrement is evaluated before *.

count=0+(9-0)=9;//b is decremented to a+3; b now points to 7.
For i=1:
count=count+(*b---i);//post decrement is evaluated before *.

count=9+(7-1)=15;//b is decremented to a+2; b now points to 5.

For i=2:
count=count+(*b---i);//post decrement is evaluated before *.

count=15+(5-2)=18;//b is decremented to a+1; b now points to 3.

Final value of count=18.

5. (b)

ptr+=sizeof(2*a[0]);
a[0] is 1. So, ptr+=sizeof(2);

Now 2 is an integer.So, ptr+=4; ptr points to 5. *(ptr-2)=3 is printed.

6. (a)

P: INCORRECT. NULL pointer dereferencing is not allowed.

Q: INCORRECT. p is an uninitialized pointer.

7. (d)

100	102	104
3	~~2~~ 3	1

	200
p	~~100~~ 102

For i=0:
*p=3;
For i=1:
*(p+1)=3-1=2;
For i=2:
*(p+2)=3-2=1;
q=100;

printf("%u\t",*++p);//*102 i.e 2 is printed.

printf("%u\t", ++*p);// ++*102 i.e 3 is printed

printf("%u\t", p-q); //(102-100)/2 i.e 1 is printed.

Output: 2 3 1

8. (c)

```
            main()
 p [ 100 ] q [ 200 ] k [ 97 ] b [ C        ]
                        100        200
printf("%d", *(char*)p-*(char *)q);
// 97-67=30
```

9. (a)

ptr is a pointer the function func().
func(a, 5):
p stores the starting address or the address of the 0th element of the array. p=a, n=5;
For i=1:
sum=sum+*(p+1)+*p++;//Post decrement operator will be evaluated before *.
sum=2+1+7;//p then points to 1.
sum=10;
For i=2:
sum=sum+*(p+2)+*p++;//Post decrement operator will be evaluated before *.
sum=10+5+1;//p then points to 3.
sum=16;
Output: 16

10. (b)

P and R returns the address of local variable. So, P and R are invalid.
Q returns the address of static variable. It is allowed.

11. (8)

func(a+2):
p stores the address of a[2].
++*p++;a[2] is incremented by 1.a[2]=4. p now points to a[3].
++*++p; p now points to a[4]. a[4] is incremented by 1.a[4]=2.

a[2]+a[3]+a[4]=4+2+2=8.

12. (a)

f(a, 3):
ptr is the pointer to the elements of a[0].
f(a+1, 2) is called.
f(a, 3) returns 4+a=4+1=5 to main().**

f(a+1, 2):
ptr is a pointer to the elements of a[1].
f(a+2, 1) is called. It returns 0.
f(a+1, 2) returns 0+(a+1) i.e. 0+4=4 to f(a, 3).**

13. (25)

((*a+1)+1) is the 1st element in the 1st 1D array of the 0th 2D array.
((*a+1)+1)+1=4+1=5
a[1][1][1] is the 1st element in the 1st 1D array of the 1st 2D array i.e 10
***(a+1) is the 0th element in the 0th 1D array of the 1st 2D array i.e 6

**(*a+1) is the 0th element in the 1st 1D array of the 0th 2D array i.e 3

*(**a+1) is the 1st element in the 0th 1D array of the 0th 2D array i.e 1

Output: 5 10 6 3 1
Sum: 25