CS & IT ENGINEERING

Programming in C

Arrays and Pointers

DPP 04 Discussion Notes

By- Pankaj Sharma sir

01 Question

02 Discussion

Q.1

[MCQ]

Consider the following codes:

P: void *p;

p=malloc(1);

*p=65;

printf("%c",*(char*)p);

Q: void *p;

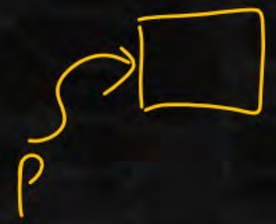
char a='A';

p=malloc(1);

p=&a;

printf("%c",*(char*)p);

Which of the following is CORRECT?



Error

A. Both P and Q are valid.

B. Only P is valid.

C. Only Q is valid.

D. Neither P nor Q is valid.

Q.2

```
#include <stdio.h>
#include <stdlib.h>
int * f()
{
   int *p=(int*)malloc(sizeof(int));
   *p=10;
   return p;
}
   int * g(int a)
{
   return &a;
}
   int main()
{
    printf("%p", f());//line 1
    printf("%p", g(15));//line 2
    return 0;
}
```

[MSQ]

A. Line 1 will result into compilation error. (Incorrect)

B. Line 2 will result into runtime error. (correct)

C. The outputs are garbage values. (Incorrect)

D. The hexadecimal addresses of pointer Variables p and local variable are displayed. (Incorrect)

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

p valid

Runtime Error

Runtime Error

A, C, D

Q.3

[MCQ]

```
#include <stdio.h>
    int main()
    {
        void *p, *q;
        int a=324;
        p=&a;
        printf("%d", *(char*)p);
        return 0;
    }
```

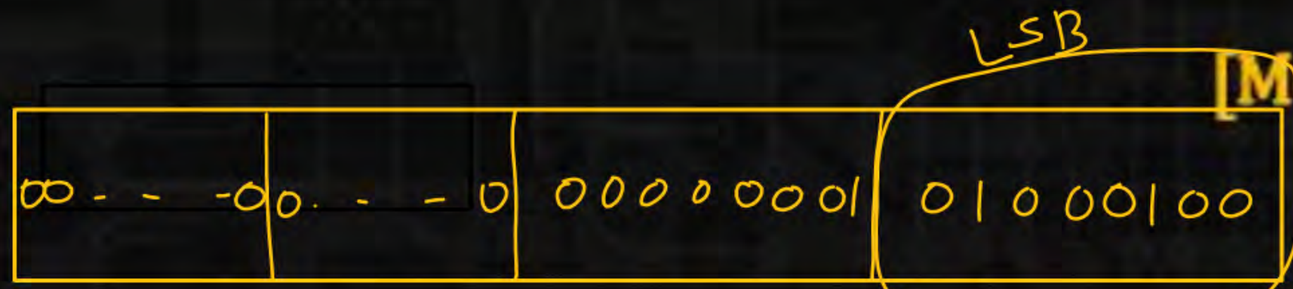
The output is-

A. Garbage value

B. Compilation error

C. 68

D. 324



01000100

⇒ 68

324

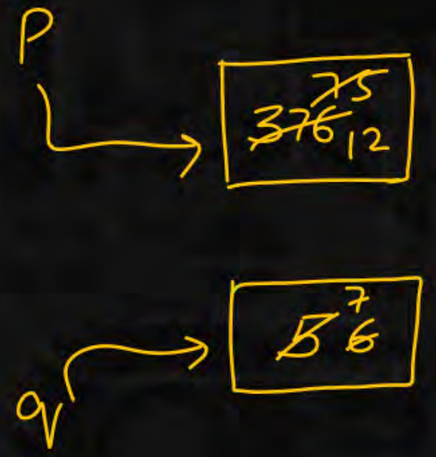
256 + 68

256 + 64 + 4

Q.4 [NAT]

```
#include <stdio.h>
#include <stdlib.h>
int main()
   {
      int *p=(int*)malloc(sizeof(int));
      int *q=(int*)malloc(sizeof(int));
      *p=376;
      *q=5;
    while(*p>*q){
          printf("%d\t",*p);
     *p/=*q;
     *q+=1;
    }
      return 0;
    }
```

The sum of the printed values is 463.



(i) $*p > *q$ $376 > 5 \Rightarrow$ true

bf $\Rightarrow$ (376)

$*p = \frac{*p}{*q} = \frac{376}{5} = 75$

$*q = *q + 1 = 5 + 1 = 6$

(ii) $*p > *q \Rightarrow 75 > 6 \Rightarrow$ true

bf $\Rightarrow$ (75)

$*p = \frac{*p}{*q} = \frac{75}{6} = (12)$

$*q = 7$

(iii) $*p > *q$

$12 > 7$

bf

$*p \Rightarrow (12)$

$*p = \frac{12}{7} = 1$

$*q = 8$

376 ✓

75 ✓

12 ✓

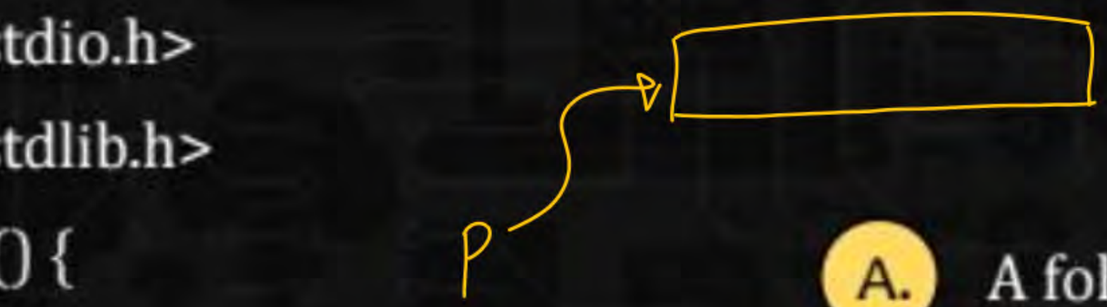
463

Q.5

[MCQ]

```
#include <stdio.h>
#include <stdlib.h>
   int main() {
   int count=0;
   char *p=(char *)malloc(sizeof(char));
   *p=65;
   printf("%c",*p);
   p=realloc(p, 4*sizeof(char));
   *p=256;
   printf("%d",*(int*)p);
   return 0;
   }
```

The output printed is-



A. A followed by Garbage values

B. A0

C. A512

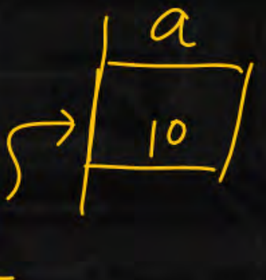
D. Compilation error

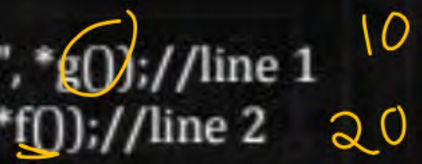
Q.6

[MCQ]

```
#include <stdio.h>
#include <stdlib.h>
    int * f()
    {
       int *p=(int*)malloc(sizeof(int));
       *p=20;
       return p;
    }
    int * g()
    {
       static int a=10;
       int *q;
       q=&a;
       return q;
    }
    int main()
    {
       printf("%d\t", *g());//line 1
       printf("%d", *f());//line 2
       return 0;
    }
```

The output is-





A. Garbage value

B. Compilation error

C. 10 20

D. 20 10

Q.7

When the memory is full, malloc returns-

[MCQ]

A. Void pointer

B. Wild pointer

C. Dangling pointer

D. NULL pointer

Q.8

```
#include <stdio.h>
#include <stdlib.h>
    int main()
    {
       int *p=(int *)calloc(2, sizeof(int));
       int *q;
       q=p+1;
       printf("%d\t", *p);
       printf("%d\t", *q);
       *p=10;
       *q=15;
       printf("%d\t", *p);
       printf("%d\t", *q);
       free(p);
       return 0;
    }
```

The output is:

[MCQ]

A. 10 15 Garbage 15

B. Garbage Garbage 10 15

C. 0 0 10 15

D. 10 15 0 0


THANK YOU GW
SOLDIERS !