

Programming in C

Dynamic Memory Allocation

DPP-09

[NAT]

1.

```
#include <stdio.h>
#include <stdlib.h>
int * f(){
    int *p=(int*)malloc(sizeof(int));
    *p=10;
    return p;
}
int main() {
    printf("%d", *f());
    return 0;
}
```

The output is _____

[MCQ]

2. 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?

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

[MSQ]

3.

```
#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;
}
```

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

- (a) Line 1 will result into compilation error.
- (b) Line 2 will result into compilation error.
- (c) The outputs are garbage values.
- (d) The hexadecimal addresses of pointer variables p and local variable are displayed.

[MCQ]

4.

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

The output is-

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

[NAT]

5.

```
#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 _____.

[MCQ]

6.

```

#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

[MCQ]

7. 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.

[MCQ]

8.

```

#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

[MCQ]

9. When the memory is full, malloc returns-

- (a) Void pointer
- (b) Wild pointer
- (c) Dangling pointer
- (d) NULL pointer

[MCQ]

10.

```

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

```

The output is:

- (a) 10 15 Garbage 15
- (b) 10 15 Garbage Garbage
- (c) 10 15 0 15
- (d) 10 15 0 0

Answer Key

- | | |
|--------------|---------|
| 1. (10) | 6. (b) |
| 2. (c) | 7. (a) |
| 3. (a, c, d) | 8. (c) |
| 4. (c) | 9. (d) |
| 5. (463) | 10. (b) |



Hints and solutions

1. (10)

Here, `malloc()` allocates bytes equivalent to that of an integer and returns the address of the allocated memory as a void pointer. The void pointer is type casted to integer type pointer 'p'.

f() p 100 return p;	Heap: 10 100
main() printf("%d", *100); //10	

2. (c)

P:

```
void *p;
p=malloc(1);
*p=65; //Invalid use of void expression. Void pointer
stores the address of any variable and needs proper
typecasting.
printf("%c",*(char*)p);
```

Hence, P is **invalid**.

Q: void *p;

```
char a='A';
p=malloc(1);
p=&a; //Void pointer p is storing the address of char
variable 'a'.
printf("%c",*(char*)p); //Proper type casting is done
hence, the code Q is valid.
```

Output : 'A'.

3. (a, c, d)

%p is the format specifier for hexadecimal memory address. The function `g()` is returning the address of local variable 'a' which will go out of scope as soon as `g()` finishes execution. So, Line 2 in the `main()` given as-
`printf("%p", g(15));`
 will give ERROR.

4.(c)

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

5. (463)

```
*p=376
*q=5
while(376>5)→True
*p=376→printf() executed.
*p=376/5=75
*q=5+1=6
while(75>6)→True
*p=75→printf() executed.
*p=75/6=12
*q=6+1=7
while(12>7)→True
*p=12→printf() executed.
*p=12/7=1
*q=7+1=8
while(1>8)→False
STOP.
```

Output: 376 75 12

Sum: 463

6. (b)

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    char *p=(char *)malloc(sizeof(char));
    *p=65;
    printf("%c",*p); //A
    p=realloc(p, 4*sizeof(char)); //realloc reallocates the
memory with size equivalent to 4 bytes and stores the
address in pointer variable 'p'.
    *p=256;
```

```
printf("%d",*(int*)p); // This will result into overflow.
return 0;
}
```

Output: A0

7. (a)

P: `int *p=NULL;`
`printf("%d", *p);`
 P is invalid NULL pointer dereferencing is not allowed.

Q: `int *p;`
`*p=10;`
`printf("%d", *p);`
 Q will result into segmentation fault. Q is also invalid.

8. (c)

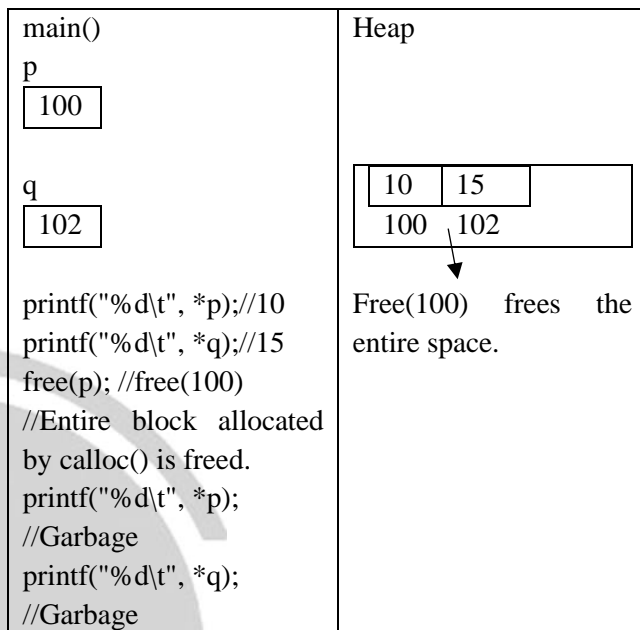
A static variable has scope throughout the program.
 The function `g()` is returning the address of static variable 'a'.
`printf("%d\t", *g());//10`
`printf("%d", *f());//20`
 Output: 10 20

9. (d)

When the memory is full, `malloc()` returns NULL pointer.

10. (b)

`Calloc()` allocates two continuous memory block of the sizes equivalent to store an integer. It returns the address to integer pointer `p`.



For more questions, kindly visit the library section: Link for app: <https://physicswallah.live/tabs/tabs/library-tab>

For more questions, kindly visit the library section: Link for web: <https://links.physicswallah.live/vyJw>

Any issue with DPP, please report by clicking here- <https://forms.gle/t2SzQVvQcs638c4r5>



PW Mobile APP: <https://play.google.com/store/apps/details?id=xyz.penpencil.physicswala>

For PW Website: <https://www.physicswallah.live/contact-us>