**Formatted Input**

**Writer:** Sayak Haldar

**1)What is the output of this C code?**

1.#include <stdio.h>

2.int main()

3.{

1. int n;
2. scanf("%d", n);
3. printf("%d\n", n);
4. return 0;

8.}

a)Compilation error

b)undefined behaviour

c)whatever user types

d)depends on the standard

**Answer is b) Undefined behaviour.**

Explanation: Compiler would throw a warning during the time of compilation stating that the argument “%d” in scanf expects argument of type int \*, but argument 2 is of type int

So, n will contain garbage value irrespective of the value which is given as input

**2)What is the output of this C code?**

1.#include <stdio.h>

2.int main()

3.{

1. char \*n;
2. scanf("%s", n);
3. return 0;

7.}

a)Compilation error

b)undefined behaviour

c)Nothing

d)none of the mentioned

**Answer is b) Undefined behaviour.**

Explanation: A character pointer can only points to the starting location of a string, if the string is initialized to the character pointer during the time of declaration.

So, in this problem, the character pointer n will point to nothing (NULL) even after scanf

**3)What is the output of this C code?**

1.#include <stdio.h>

2.int main()

3.{

4. char n[] = "hello\nworld!";

5. char s[13];

6. sscanf(n, "%s", s);

7. printf("%s\n", s);

8. return 0;

9.}

a)hellonworld!

b)hello

world!

c)hello

d)hello world!

**Answer is c) Hello**

Explanation: sscanf exactly behaves like scanf except that it reads formatted input from a string

instead of stream attached to stdin

So, '\n' acts as a delimiter

**4)What is the output of this C code?**

1.#include <stdio.h>

2.int main()

3.{

4. short int i;

5. scanf("%hd", &i);

6. printf("%hd", i);

7. return 0;

8.}

a)Compilation error

b)Undefined behaviour

c)Whatever user types

d)None of the mentioned

**Answer is c) Whatever user types**

Explanation: The h format specifier is used alongside with d for short int

**5)What is the output of this C code?**

1.#include <stdio.h>

2.int main()

3.{

4. short int i;

5. scanf("%\*d", &i);

6. printf("%hd", i);

7. return 0;

8.}

a)Compilation error

b)Somegarbage value

c)Whatever user types

d)Depends on the standard

**Answer) b) Some garbage value.  
  
Explanation:** Due to the presence of \* in scanf(“%\*d”,&i) the first value given in stdin will be skipped. So, some garbage value will be printed

**6)What is the output of this C code?**

1.#include <stdio.h>

2.int main()

3.{

1. short int i;
2. scanf("%\*hd", &i);
3. printf("%hd", i);
4. return 0;

8.}

a)Compilation error

b)Somegarbage value

c)Whatever user types

d)Depends on the standard

**Answer is b) Somegarbagevalue**

Explanation: Due to the presence of \* in scanf(“%\*hd”,&i) the first value given in stdin will be skipped. So, some garbage value will be printed.

**7)What is the output of this C code? Consider input as a large number whose value exceeds the maximum range of short int**

1.#include <stdio.h>

2.int main()

3.{

4. short int i;

5. scanf("%hd", &i);

6. printf("%hd", i);

7. return 0;

8.}

a)Compilation error

b)Somegarbage value

c)Whatever user types

d)Depends on the standard

**Answer is b) Somegarbage value**

Explanation: Actually, some negative value will be printed due to the cyclic property of primitive datatypes in c

**8)What is the output of this C code?**

1.#include <stdio.h>

2.int main()

3.{

4.short int i;

5.scanf("%h\*d", &i);

6.printf("%hd", i);

7.return 0;

8.}

a)Compilation error

b)Undefined behaviour

c)Somegarbage value

d)Depends on the standard.

**Answer is a) Compilation error Explanation: h\*d is not a valid format specifier.**

**9) Which of the following is NOT a delimiter for an input in scanf?**

a)Enter

b)Space

c)Tab

d) None of the mentioned

**Answer) d) None of the mentioned.**

**10)If the conversion characters of int d, i, o, u and x are preceded by h, it indicates?**

a)A pointer to int

b)A pointer to short

c)A pointer to long

d)A pointer to char

**Answer is b) A pointer to short**

**11) Which of the following doesn’t require an & for the input in scanf?**

a)char name[10];

b)int name[10];

c)float name[10];

d)All of the mentioned

**Answer) is a)**

**12) Which of the following is an invalid method for input?**

a)scanf(“%d%d%d”,&a, &b, &c);

b)scanf(“%d %d %d”, &a, &b, &c);

c)scanf(“Three values are %d %d %d”,&a,&b,&c);

d)None of the mentioned.

**Answer is d) None of the mentioned. Not even c)**

However, you can’t see the words “Three values are” and the values.

**13) Which of the following represents the function for scanf?**

a)void scanf(char \*format, …)

b)int scanf(char \*format, …)

c)char scanf(int format, …)

d)char \*scanf(char \*format, …)

**Answer is b) int scanf(char \*format,....)**

It returns number of successfully matches and assigned input items.

**14) scanf returns as its value**

a)Number of successfully matched and assigned input items

b)Nothing

c)Number of characters properly printed

d)Error

**Answer) a)**

**15)What is the output of this C code?**

1.#include <stdio.h>

2.void main()

3.{

1. int n;
2. scanf("%d", n);
3. printf("%d", n);
4. return 0;

8.}

a)Prints the number that was entered

b)Segmentation fault

c)Nothing

d)Varies

**Answer) d) Varies.**

Compiler would throw a warning for line 5, saying that “%d” in scanf expects an argument of type int \*, but argument 2 is of type int. It will print a garbage value and that value varies.

**16) int sscanf(char \*string, char \*format, arg1, arg2, …)**

a)Scans the string according to the format in format and stores the resulting values through arg1, arg2, etc.

b)The arguments arg1,arg2 etc must be pointers

c)Both a & b

d)None of the mentioned

**Answer is c) Both a & b**

**17)The conversion characters d, i, o, u, and x may be preceded by h in scanf to indicate**

a)A pointer to short

b)A pointer to long

c)Nothing

d)Error

**Answer) is a)**

**18)What is the output of this C code (when 4 and 5 are entered)?**

1.#include <stdio.h>

2.int main()

3.{

1. int m, n;
2. printf("enter a number");
3. scanf("%d", &n);
4. scanf("%d", &m);
5. printf("%d\t%d\n", n, m);
6. return 0;

10.}

1. Error
2. 4 junkvalue
3. Junkvalue 5
4. 4 5

**Answer is d) 4 5**