**Constants**

**Writer:** Sayak Haldar

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

#include <stdio.h>

int main()

{

enum {ORANGE = 5, MANGO, BANANA = 4, PEACH};

printf("PEACH = %d\n", PEACH);

}

a) PEACH = 3

b) PEACH = 4

c) PEACH = 5

d) PEACH = 6

**Answer: c**

Explanation: In enum, the value of constant is defined to the recent assignment from left.

**Output:**

$ cc pgm1.c

$ a.out

PEACH = 5

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

#include <stdio.h>

int main()

{

printf("C programming %s", "Class by\n%s Sanfoundry", "WOW");

}

a) C programming Class by

WOW Sanfoundry

b) C programming Class by\n%s Sanfoundry

c) C programming Class by

%s Sanfoundry

d) Compilation error

**Answer: c**

**Explanation:** This program has only one %s within first double quotes, so it does not read the string “WOW”.

The %s along with the Sanfoundry is not read as a format modifier while new line character prints the new line.

**Output:**

$ cc pgm2.c

$ a.out

C programming Class by

%s Sanfoundry

**3. For the following code snippet:**

char \*str = “Sanfoundry.com\0” “training classes”;

The character pointer str holds reference to string:

a) Sanfoundry.com

b) Sanfoundry.com\0training classes

c) Sanfoundry.comtraining classes

d) Invalid declaration

**Answer: b**

Explanation: ‘\0’ is accepted as a char in the string. Even though strlen will give length of string “Sanfoundry.com”, in memory str is pointing to entire string including training classes”

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

#include <stdio.h>

#define a 10

int main()

{

const int a = 5;

printf("a = %d\n", a);

}

a) a = 5

b) a = 10

c) Compilation error

d) Runtime error

View Answer

**Answer: c**

Explanation: The #define substitutes a with 10 leaving no identifier and hence compilation error.

Output:

$ cc pgm3.c

pgm3.c: In function ‘main’:

pgm3.c:5: error: expected identifier or ‘(’ before numeric constant

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

#include <stdio.h>

int main()

{

int var = 010;

printf("%d", var);

}

a) 2

b) 8

c) 9

d) 10

**Answer: b**

Explanation: 010 is octal representation of 8.

Output:

$ cc pgm4.c

$ a.out

8

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

#include <stdio.h>

enum birds {SPARROW, PEACOCK, PARROT};

enum animals {TIGER = 8, LION, RABBIT, ZEBRA};

int main()

{

enum birds m = TIGER;

int k;

k = m;

printf("%d\n", k);

return 0;

}

a) 0

b) Compile time error

c) 1

d) 8

**Answer: d**

Explanation: m is an integer constant, hence compatible.

**Output:**

8

**7. What is the output of this C code?**

#include <stdio.h>

#define MAX 2

enum bird {SPARROW = MAX + 1, PARROT = SPARROW + MAX};

int main()

{

enum bird b = PARROT;

printf("%d\n", b);

return 0;

}

a) Compilation error

b) 5

c) Undefined value

d) 2

**Answer: b**

Explanation: MAX value is 2 and hence PARROT will have value 3 + 2.

Output:

$ cc pgm6.c

$ a.out

5

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

#include <stdio.h>

#include <string.h>

int main()

{

char \*str = "x";

char c = 'x';

char ary[1];

ary[0] = c;

printf("%d %d", strlen(str), strlen(ary));

return 0;

}

a) 1 1

b) 2 1

c) 2 2

d) 1 (undefined value)

**Answer: d**

Explanation: str is null terminated but ary is not.

Output:

$ cc pgm7.c

$ a.out

1 5

**9. enum types are processed by**

a) Compiler

b) Preprocessor

c) Linker

d) Assembler

**Answer) a) Compiler.**

**10. What is the output of this C code?**

#include <stdio.h>

int main()

{

printf("sanfoundry\rclass\n");

return 0;

}

a) sanfoundryclass

b) sanfoundry

class

c) classundry

d) sanfoundry

**Answer) c**

Explanation: r is carriage return and moves the cursor back. sanfo is replaced by class

Output:

$ cc pgm8.c

$ a.out

classundry

**11. What is the output of this C code?**

#include <stdio.h>

int main()

{

printf("sanfoundry\r\nclass\n");

return 0;

}

a) sanfoundryclass

b) sanfoundry

class

c) classundry

d) sanfoundry

**Answer: b**

Explanation: rn combination makes cursor move to nextline.

Output:

$ cc pgm9.c

$ a.out

sanfoundry

class

**12. What is the output of this C code?**

#include <stdio.h>

int main()

{

const int p;

p = 4;

printf("p is %d", p);

return 0;

}

a) p is 4

b) Compile time error

c) Run time error

d) p is followed by a garbage value

**Answer) b) Compile time error.**

const is a type qualifier which is applied to int datatype resulting qualified type.  
Now, since the constant variable has to be declared and defined at the same time, not doing it results in an error.

**13. Comment on the output of this C code?**

#include <stdio.h>

void main()

{

int k = 4;

int \*const p = &k;

int r = 3;

p = &r;

printf("%d", p);

}

a) Address of k

b) Address of r

c) Compile time error

d) Address of k + address of r

**Answer: c**

Explanation: Since the pointer p is declared to be constant, trying to assign it with a new value results in an error.

**14. Which is false?**

a) Constant variables need not be defined as they are declared and can be defined later

b) Global constant variables are initialised to zero

c) const keyword is used to define constant values

d) You cannot reassign a value to a constant variable

**Answer) a)**

**15. Comment on the output of this C code?**

#include <stdio.h>

void main()

{

int const k = 5;

k++;

printf("k is %d", k);

}

a) k is 6

b) Error due to const succeeding int

c) Error, because a constant variable can be changed only twice

d) Error, because a constant variable cannot be changed

**Answer: d**

Explanation: Constant variable has to be declared and defined at the same time. Trying to change it results in an error.

Output:

$ cc pgm12.c

pgm12.c: In function ‘main’:

pgm12.c:5: error: increment of read-only variable ‘k’

**16. Comment on the output of this C code?**

#include <stdio.h>

int const print()

{

printf("Sanfoundry.com");

return 0;

}

void main()

{

print();

}

a) Error because function name cannot be preceded by const

b) Sanfoundry.com

c) Sanfoundry.com is printed infinite times

d) Blank screen, no output

**Answer: b**

#include <stdio.h>

int const print()

{

printf("Sanfoundry.com\n");

return 0;

}

int main()

{

int ret\_val=print();

printf("The ret val is %d\n",ret\_val);

ret\_val=1;

return 0;

}

Now, note that, this will not generate compilation error. Because, the read only value returned by print was copied to ret\_val and ret\_val is being changed.