C Programming
String in C programming

DPP:01

Q1 Consider the following codes P and Q as:

P : char* p ="GATEWallah";
p[5]= 'A';
printf("%s",p);

Q: char* p ="GATEWallah";
char* q = p;
q[5]= 'A';
printf("%s",q);

The number of INCORRECT codes is/are _____.

Q2
```
P : char s1[]="GATE";
    char s2[]="GATE";
    if(s1==s2) printf("YES");
    else
    printf("NO");
Q: char s1[]="GATE";
    char s2[]="GateWallah";
    if(*s1==*s2) printf("YES");
    else
    printf("NO");
```

The outputs are-

(A) P = YES Q = YES
(B) P = YES Q = NO
(C) P = NO Q = YES
(D) P = NO Q = NO

Q3
```
P : char  s[20];
    printf("Enter your GATE stream with year: \n");
    scanf("%s",s);
    printf("%s",s);
Q : char  s[20];
    printf("Enter your GATE stream with year: \n");
    gets(s);
    printf("%s",s);
```

If the input string is "CS 2023", the outputs are-

(A) P = CS 2023 Q = CS 2023
(B) P = CS Q = CS
(C) P = CS 2023 Q = CS
(D) P = CS Q = CS 2023

Q4
```
#include<stdio.h>
#include<string.h>
int main()
{
    char s[20]="GATEWallah";
    printf("%s",s+4);
    s[4]=0;
    printf("%s",s);
    return 0;
 }
```

The output is

(A) WallahGATE
(B) EWallahGAT
(C) WallahGATE0allah
(D) EWallahGAT0allah

Q5
```
#include<stdio.h>
#include<string.h>
int main()
{
    char s[20]="GATEWallah2023";
    s[10]='0';
    printf("%s",s+s[3]-s[1]);
    return 0;
 }
```

The output printed is-

(A) Wallah0 (B) Wallah2023
(C) Wallah0023 (D) Wallah

Q6
```
#include<stdio.h>
#include<string.h>
void f(char *p)
{
    static int q=2;
    q=q+3;
    p[q]+=2;
 }
int main()
{
    char s[20]="GATEWallahbesthai";
    int i=0;
    for(i=0;i<3;i++)
    {
        f(s);
     }
    printf("%s",s);
    return 0;
}
```
The output string printed is

(A) GATEWcllchbgsthai

(B) GATEWcllbhbgsthai

(C) GATEWcllchbesthai

(D) GATEWcllchbesthai

Q7
```
#include<stdio.h>
#include<string.h>
void f(char *p)
{
    if(*p!=0)
    {
        printf("%c", *p);
        f(p+1);
     }
     printf("%c", *p);
}
int main()
{
    char s[5]="GATE";
    f(s);
    return 0;
}
```
The output is

(A) GATEGATE (B) ETAGGATE

(C) ETAGETAG (D) GATEETAG

Q8
```
#include<stdio.h>
#include<string.h>
int main()
{
    int a=1;
    char b[]="GATE2024";
    char c[]="GATE2024";
    int d=strcmp(b,c);
    if(d==0)
    a=printf("GATEWallah");
    printf("%d",a);
    return 0;
 }
```
The value of a is _____.

Answer Key

Q1	2~0	Q5	(C)
Q2	(C)	Q6	(A)
Q3	(D)	Q7	(D)
Q4	(A)	Q8	10~9

Hints & Solutions

Q1 Text Solution:

char*p = "GATEWallah";

Memory is allocated to "GATEWallah" in static/ read only memory. So, its content cannot be updated

p[5] = 'A'

It is not allowed as 'p' is the only entry point to the string constant.

\ Both P and Q are not valid.

Q2 Text Solution:

P: if (s1 = = s2) // It is comparing the base addresses of two different Strings.

→ false

∴ else part will be executed

⇓

No is printed

Q:

s1: GATE (100)

s2: GATEWalla h (200)

if(*s1 = = *s2) ⇒ if (*100 = = *200)

* → value at

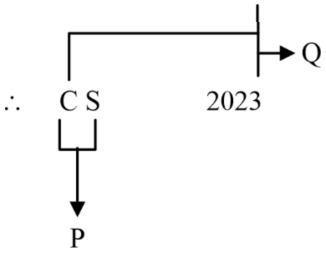
⇓

G = = G

→ TRUE

Q3 Text Solution:

scanf() halts reading as soon as it encounters whitespace. gets() ignores the whitespace and stops reading when new-line is found.



∴ Output of P: CS

Output of Q: CS 2023

Q4 Text Solution:

	100	101	102	103	104	105	106	107	108	109	110
S:	G	A	T	E	~~W~~ \0	a	l	l	a	h	\0

printf("%s", s + 4); // Wallah

↓

104

s[4] = 0; //*(100 +4) = 0 where 0 is the ASCII of NULL character.

print("%s", s); // It prints the string till it encounters first NULL;

⇒Output is: WallahGATE

Q5 Text Solution:

	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114
S:	G	A	T	E	W	a	l	l	a	h	~~2~~ 0	0	2	3	\0

s[10] = '0'; // Here '0' is the numeri 0

printf("%s", s+s[3]-s[1]);

⇓

100 + 69 − 65 = 104

∴Output is: Wallah0023

Q6 Text Solution:

S:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
G	A	T	E	W	a	l	l	a	h	b	e	s	t	h	a	i

starting address of S: 100

i 0

f(100)

p 100

q ~~2~~ 5

p[5] + = 2; // p[5] = c

i 1

f(100)

p 100

q ~~8~~ 8

p[8] + = 2; // p[8] = c

i 2

f(100)

p 100

q ~~8~~ 11

p[11] + = 2; // p[11] = g

Output: G A T E W c l l c h b g s t h a i

Q7 Text Solution:

G	A	T	E	\0
100	101	102	103	104

f(100) *100==G!=0→True **(1)** printf( ) executed → G f(101) **(8)** printf( ) executed → G	f(101) *101==A!=0→True **(2)** printf( ) executed → A f(102) **(7)** printf( ) executed → A
f(102) *102==T!=0→True **(3)** printf( )executed → T f(103) **(6)** printf( ) executed → T	f(103) *103==E!=0→True **(4)** printf( ) executed →E f(104)→NULL is present **(5)** printf( )executed→E

∴ Output is: GATEETAG