STRINGS

1. Assume sizeof an integer and a pointer is 4 byte. Output?

```
#include <stdio.h>
#define R 10
#define C 20

int main()
{
   int (*p)[R][C];
   printf("%d",  sizeof(*p));
   getchar();
return 0;
}
```

(a) 200
(b) 4
(c) 800
(d) 80

2. Output of following C program? Assume that all necessary header files are included.

```
int main()
{
  char *s1 = (char *)malloc(50);
  char *s2 = (char *)malloc(50);
  strcpy(s1, "Ravindra");
  strcpy(s2, "Ravula");
  strcat(s1, s2);
  printf("%s", s1);
  return 0;
}
```

(a) RavindraRavula
(b) Ravindra
(c) Ravindra Ravula
(d) Ravula

3. Consider the following code. The function myStrcat concatenates two strings. It appends all characters of b to end of a. So the expected output is “Ravindra Ravula”. The program compiles fine but produces segmentation fault when run.

```
#include <stdio.h>

#include<string.h>

void myStrcat(char *a, char *b)
{
   int m = strlen(a);
   int n = strlen(b);
   int i;
   for (i = 0; i <= n; i++)
     a[m+i]  = b[i];
}

int main()
{
   char *str1 = "Ravindra ";
   char *str2 = "Ravula";
   myStrcat(str1, str2);
   printf("%s ", str1);
   return 0;
}
```

Which of the following changes can correct the program so that it prints “Geeks Quiz”?

(a) char *str1 = “Ravindra “; can be changed to char str1[100] = “Ravindra “;
(b) char *str1 = “Ravindra “; can be changed to char str1[100] = “Ravindra “; and a line a[m+n- 1] = ‘\0′ is added at the end of myStrcat
(c) A line a[m+n-1] = ‘\0′ is added at the end of myStrcat
(d) A line ‘a = (char *)malloc(sizeof(char)*(strlen(a) + strlen(b) + 1)) is added at the beginning of myStrcat()

4. What is the output of following program?

```
# include <stdio.h>

int main()
{
  char str1[] = "Ravindras";
  char str2[] = {'R', 'a', 'v', 'i', 'n', 'd', 'r', 'a', 's'};
  int n1 = sizeof(str1)/sizeof(str1[0]);
```

```
  int n2 = sizeof(str2)/sizeof(str2[0]);
  printf("n1 = %d, n2 = %d", n1, n2);
  return 0;
}
```

(a) n1 = 10, n2 = 9
(b) n1 = 10, n2 = 10
(c) n1 = 9, n2 = 9
(d) n1 = 9, n2 = 10

5. What is the output of following program?

```
#include<stdio.h>

void swap(char *str1, char *str2)
{
  char *temp = str1;
  str1 = str2;
  str2 = temp;
}

int main()
{
  char *str1 = "Ravindra";
  char *str2 = "ravula";
  swap(str1, str2);
  printf("str1 is %s, str2 is %s", str1, str2);
  return 0;
}
```

(a) str1 is ravula, str2 is Ravindra
(b) str1 is Ravindra, str2 is ravula
(c) str1 is Ravindra, str2 is Ravindra
(d) str1 is ravula, str2 is ravula

6. Predict the output?

```
#include <stdio.h>

int fun(char *str1)
{
  char *str2 = str1;
  while(*++str1);
```

```
  return (str1-str2);
}

int main()
{
  char *str = "Ravindrar";
  printf("%d", fun(str));
  return 0;
}
```

(a) 10
(b) 9
(c) 8
(d) Random Number

7. What does the following fragment of C-program print?

```
char c[] = "GATE2011";
char *p =c;
printf("%s", p + p[3] - p[1]) ;
```

(a) GATE2011
(b) E2011
(c) 2011
(d) 011

8.

```
#include<stdio.h>

int main()
{
   char str[] = "ravindrar";
   printf("%s %s %s\n", &str[5], &5[str], str+5);
   printf("%c %c %c\n", *(str+6), str[6], 6[str]);
   return 0;
}
```

(a) Runtime Error
(b) Compiler Error
(c) rar rar rar
    r r r
(d) drar drar drar
    u u u

9. In below program, what would you put in place of "?" to print "Quiz"?

```
#include <stdio.h>

int main()
{
  char arr[] = "GatesQuiz";
  printf("%s", ?);
  return 0;
}
```

(a) arr
(b) (arr+5)
(c) (arr+4)
(d) Not possible

10. Output?

```
int main()
{
   char a[2][3][3] = {'g','a','t','e','s','q','u','i','z'};
   printf("%s ", **a);
   return 0;
}
```

(a) Compiler Error
(b) gatesquiz followed by garbage characters
(c) gatesquiz
(d) Runtime Error

11. Consider the following C program segment:

```
char p[20];
char *s = "string";
int length = strlen(s);
int i;
for (i = 0; i < length; i++)
   p[i] = s[length — i];
printf("%s", p);
```

The output of the program is? (GATE CS 2004)

(a) gnirts
(b) gnirt

(c) string
(d) no output is printed

12.

```
#include <stdio.h>

void my_toUpper(char* str, int index)
{
   *(str + index) &= ~32;
}

int main()
{
   char* arr = "gatesquiz";
   my_toUpper(arr, 0);
   my_toUpper(arr, 5);
   printf("%s", arr);
   return 0;
}
```

(a) GatesQuiz
(b) gatesquiz
(c) Compiler dependent

13. Predict the output of the following program:

```
#include <stdio.h>

int main()
{
char str[] = "%d %c", arr[] = "GatesQuiz";
printf(str, 0[arr], 2[arr + 3]);
return 0;
}
```

(a) G Q
(b) 71 81
(c) 71 Q
(d) Compile-time error

14. Output of following program

```
#include <stdio.h>

int fun(char *p)
{
   if (p == NULL || *p == '\0') return 0;
   int current = 1, i = 1;
   while (*(p+current))
  {
     if (p[current] != p[current-1])
     {
         p[i] = p[current];
         i++;
      }
      current++;
   }
   *(p+i)='\0';
   return i;
}

int main()
{
   char str[] = "geeksskeeg";
   fun(str);
   puts(str);
   return 0;
}
```

(a) gekskeg
(b) geeksskeeg
(c) geeks
(d) Garbage Values

15.

```
int main()
{
   char p[] = "gatesquiz";
   char t;
   int i, j;
   for(i=0,j=strlen(p); i<j; i++)
   {
```

```
        t = p[i];
        p[i] = p[j-i];
        p[j-i] = t;
     }
   printf("%s", p);
   return 0;
}
```

Output?

(a) ziuqsetag
(b) Nothing is printed on the screen
(c) gatesquiz
(d) gggggggg

16. Assume that a character takes 1 byte. Output of following program?

```
#include<stdio.h>

int main()
{
   char str[20] = "GatesQuiz";
   printf ("%d", sizeof(str));
   return 0;
}
```

(a) 9
(b) 10
(c) 20
(d) Garbage Value

17. Predict the output of following program, assume that a character takes 1 byte and pointer takes 4 bytes.

```
#include <stdio.h>

int main()
{
   char *str1 = "GatesQuiz";
   char str2[] = "GatesQuiz";
   printf("sizeof(str1) = %d, sizeof(str2) = %d",
         sizeof(str1), sizeof(str2));
```

```
    return 0;
}
```

(a) sizeof(str1) = 10, sizeof(str2) = 10
(b) sizeof(str1) = 4, sizeof(str2) = 10
(c) sizeof(str1) = 4, sizeof(str2) = 4
(d) sizeof(str1) = 10, sizeof(str2) = 4

18. The output of following C program is

```
#include <stdio.h>

char str1[100];
char *fun(char str[])
{
    static int i = 0;
    if (*str)
        {
        fun(str+1);
        str1[i] = *str;
        i++;
         }
    return str1;
}

int main()
{
   char str[] = "GATE CS 2016 Mock Test";
    printf("%s", fun(str));
    return 0;
}
```

(a) GATE CS 2016 Mock Test
(b) tseT kcoM 6102 SC ETAG
(c) Nothing is printed on screen
(d) Segmentation Fault

19. Consider the following function written in the C programming language.

The output of the above function on input “ABCD EFGH” is

```
void foo (char *a)
{
```

```
  if (*a && *a != ` `)
      {
     foo(a+1);
     putchar(*a);
      }
}
```

(a) ABCD EFGH
(b) ABCD
(c) HGFE DCBA
(d) DCBA

20. Consider the following C program segment.

```
# include <stdio.h>

int main( )
{
   char s1[7] = "1234", *p;
   p = s1 + 2;
   *p = '\0' ;
   printf ("%s", s1);
}
```

What will be printed by the program?

(a) 12
(b) 120400
(c) 1204
(d) 1034