

# **Rajalakshmi Engineering College**

**Rajalakshmi Nagar, Thandalam, Chennai - 602 105**

## **Department of Computer Science and Engineering**



**Let Us C**

**Yashavant Kanetkar**

**Find the Output - Solutions**

**(Turbo C++)**



**B.BHUVANESWARAN**

**Assistant Professor (SG) / CSE / REC**  
**bhuvaneshwaran@rajalakshmi.edu.in**



**Chapter - 01**  
**Getting Started**

## Chapter - 02

### C Instructions

What will be the output of the following programs?

(a)

```
#include <stdio.h>
int main()
{
    int i = 2, j = 3, k, l;
    float a, b;
    k = i / j * j;
    l = j / i * i;
    a = i / j * j;
    b = j / i * i;
    printf("%d %d %f %f\n", k, l, a, b);
    return 0;
}
```

Output:

0 2 0.000000 2.000000

(b)

```
#include <stdio.h>
int main()
{
    int a, b, c, d;
    a = 2 % 5;
    b = -2 % 5;
    c = 2 % -5;
    d = -2 % -5;
    printf("a = %d b = %d c = %d d = %d\n", a, b, c, d);
    return 0;
}
```

Output:

a = 2 b = -2 c = 2 d = -2

(c)

```
# include <stdio.h>
int main()
{
    float a = 5, b = 2;
    int c, d;
    c = a % b;
    d = a / 2;
    printf("%d\n", d);
    return 0;
}
```

Output:

Error. Mod ( % ) operator cannot be used on **floats**

(d)

```
# include <stdio.h>
int main()
{
    printf("nn \n\n nn\n");
    printf("nn /n/n nn/n");
    return 0;
}
```

Output:

```
nn

nn
nn /n/n nn/n
```

(e)

```
# include <stdio.h>
int main()
{
    int a, b;
    printf("Enter values of a and b");
    scanf(" %d %d ", &a, &b);
    printf("a = %d b = %d", a, b);
    return 0;
}
```

Output:

Since spaces are given after and before double quotes in **scanf( )** we must supply a space, then two numbers and again a space followed by enter. The **printf( )** would then output the two number that you enter.

## Chapter - 03

### Decision Control Instruction

What will be the output of the following programs?

(a)

```
#include <stdio.h>
int main()
{
    int a = 300, b, c;
    if (a >= 400)
        b = 300;
    c = 200;
    printf("%d %d\n", b, c);
    return 0;
}
```

Output:

0 200

b will contain some garbage value and c will be equal to 200

(b)

```
# include <stdio.h>
int main()
{
    int a = 500, b, c;
    if (a >= 400)
        b = 300;
    printf("%d %d\n", b, c);
    return 0;
}
```

Output:

300 0

(c)

```
# include <stdio.h>
int main()
{
    int x = 10, y = 20;
    if (x == y);
    printf("%d %d\n", x, y);
    return 0;
}
```

Output:

10 20

(d)

```
#include <stdio.h>
int main()
{
    int x = 3;
    float y = 3.0;
    if (x == y)
        printf("x and y are equal\n");
    else
        printf("x and y are not equal\n");
    return 0;
}
```

Output:

x and y are equal

(e)

```
#include <stdio.h>
int main()
{
    int x = 3, y, z;
    y = x = 10;
    z = x < 10;
    printf("x = %d y = %d z = %d\n", x, y, z);
    return 0;
}
```

Output:

x = 10 y = 10 z = 0

(f)

```
# include <stdio.h>
int main()
{
    int i = 65;
    char j = 'A';
    if (i == j)
        printf("C is WOW\n");
    else
        printf("C is a headache\n");
    return 0;
}
```

*Output:*

C is WOW



## Chapter - 04

### More Complex Decision Making

What will be the output of the following programs?

(a)

```
#include <stdio.h>
int main()
{
    int i = 4, z = 12;
    if (i = 5 | z > 50)
        printf("Dean of students affairs\n");
    else
        printf("Dosa\n");
    return 0;
}
```

Output:

Dean of students affairs

(b)

```
#include <stdio.h>
int main( )
{
    int i = 4, j = -1, k = 0, w, x, y, z;
    w = i || j | k;
    x = i && j && k;
    y = i | j && k;
    z = i && j | k;
    printf("w = %d x = %d y = %d z = %d\n", w, x, y, z);
    return 0 ;
}
```

Output:

w = 1 x = 0 y = 0 z = 1

(c)

```
#include <stdio.h>
int main()
{
    int x = 20, y = 40, z = 45;
    if (x > y && x > z)
        printf("biggest = %d\n", x);
    else if (y > x && y > z)
        printf("biggest = %d\n", y);
    else if (z > x && z > y)
        printf("biggest = %d\n", z);
    return 0;
}
```

Output:

biggest = 45

(d)

```
#include <stdio.h>
int main()
{
    int i = -1, j = 1, k, l;
    k = !i && j;
    l = !i | j;
    printf("%d %d\n", i, j);
    return 0;
}
```

Output:

-1 1

(e)

```
#include <stdio.h>
int main()
{
    int i = -4, j, num;
    j = (num < 0 ? 0 : num * num);
    printf("%d\n", j);
    return 0;
}
```

Output:

Unpredictable. num not initialised

(f)

```
# include <stdio.h>
int main()
{
    int k, num = 30;
    k = (num > 5 ? (num <= 10 ? 100 : 200) : 500);
    printf("%d\n", num);
    return 0;
}
```

*Output:*

30

## Chapter - 05

### Loop Control Instruction

What will be the output of the following programs?

(a)

```
#include <stdio.h>
int main()
{
    int i = 1;
    while (i <= 10);
    {
        printf("%d\n", i);
        i++;
    }
    return 0;
}
```

*Output:*

No Output Indefinite while loop because of a ';' at the end of while

(b)

```
#include <stdio.h>
int main()
{
    int x = 4, y, z;
    y = --x;
    z = x--;
    printf("%d %d %d\n", x, y, z);
    return 0;
}
```

*Output:*

2 3 3

(c)

```
#include <stdio.h>
int main()
{
    int x = 4, y = 3, z;
    z = x-- - y;
    printf("%d %d %d\n", x, y, z);
    return 0;
}
```

Output:

3 3 1

(d)

```
#include <stdio.h>
int main()
{
    while ('a' < 'b')
        printf("malayalam is a palindrome\n");
    return 0;
}
```

Output:

'malayalam is a palindrome' will be printed indefinitely

(e)

```
#include <stdio.h>
int main()
{
    int i;
    while (i = 10)
    {
        printf("%d\n", i);
        i = i + 1;
    }
    return 0;
}
```

Output:

10 will be printed indefinitely.

(f)

```
#include <stdio.h>
int main()
{
    float x = 1.1;
    while (x == 1.1)
    {
        printf("%f\n", x);
        x = x - 0.1;
    }
    return 0;
}
```

*Output:*

No output. Since a **float** variable is compared with **double** constant, condition will not satisfy.

## Chapter - 06

### More Complex Repetitions

What will be the output of the following programs?

(a)

```
#include <stdio.h>
int main()
{
    int i = 0;
    for (; i;)
        printf("Here is some mail for you\n");
    return 0;
}
```

*Output:*

No Output

(b)

```
#include <stdio.h>
int main()
{
    int i;
    for (i = 1; i <= 5; printf("%d\n", i));
    i++;
    return 0;
}
```

*Output:*

1 will be printed indefinite number of times.

(c)

```
#include <stdio.h>
int main()
{
    int i = 1, j = 1;
    for (;;)
    {
        if (i > 5)
            break;
        else
            j += i;
        printf("%d\n", j);
        i += j;
    }
}
```

```
        return 0;  
    }
```

*Output:*

```
2  
5
```



## Chapter - 07

### Case Control Instruction

What will be the output of the following programs?

(a)

```
#include <stdio.h>
int main()
{
    char suite = 3;
    switch (suite)
    {
        case 1:
            printf("Diamond\n");
        case 2:
            printf("Spade\n");
        default:
            printf("Heart\n");
    }
    printf("I thought one wears a suite\n");
    return 0;
}
```

Output:

Heart  
I thought one wears a suite

(b)

```
#include <stdio.h>
int main()
{
    int c = 3;
    switch (c)
    {
        case '3':
            printf("You never win the silver prize\n");
            break;
        case 3:
            printf("You always lose the gold prize\n");
            break;
        default:
            printf("Of course provided you win a prize\n");
    }
    return 0;
}
```

Output:

You always lose the gold prize

(c)

```
#include <stdio.h>
int main()
{
    int i = 3;
    switch (i)
    {
        case 0:
            printf("Customers are dicey\n");
        case 1 + 0:
            printf("Markets are pricey\n");
        case 4 / 2:
            printf("Investors are moody\n");
        case 8 % 5:
            printf("At least employees are good\n");
    }
    return 0;
}
```

Output:

At least employees are good

(d)

```
#include <stdio.h>
int main()
{
    int k;
    float j = 2.0;
    switch (k = j + 1)
    {
        case 3:
            printf("Trapped\n");
            break;
        default:
            printf("Caught!\n");
    }
    return 0;
}
```

Output:

Trapped

(e)

```
#include <stdio.h>
int main()
{
    int ch = 'a' + 'b';
    switch (ch)
    {
        case 'a':
        case 'b':
            printf("You entered b\n");
        case 'A':
            printf("a as in ashar\n");
        case 'b' + 'a':
            printf("You entered a and b\n");
    }
    return 0;
}
```

Output:

You entered a and b

(f)

```
#include <stdio.h>
int main()
{
    int i = 1;
    switch (i - 2)
    {
        case -1:
            printf("Feeding fish\n");
        case 0:
            printf("Weeding grass\n");
        case 1:
            printf("Mending roof\n");
        default:
            printf("Just to survive\n");
    }
    return 0;
}
```

Output:

Feeding fish  
Weeding grass  
Mending roof  
Just to survive

## Chapter - 08

### Functions

What will be the output of the following programs?

(a)

```
#include <stdio.h>

void display();

int main()
{
    printf("Learn C\n");
    display();
    return 0;
}

void display()
{
    printf("Followed by C++, C# and Java!\n");
    main();
}
```

*Output:*

Both the messages will get printed indefinitely

(b)

```
# include <stdio.h>

int check(int);

int main()
{
    int i = 45, c;
    c = check(i);
    printf("%d\n", c);
    return 0;
}

int check(int ch)
{
    if (ch >= 45)
        return (100);
    else
        return (10 * 10);
}
```

Output:

100

(c)

```
#include <stdio.h>

float circle(int);

int main()
{
    float area;
    int radius = 1;
    area = circle(radius);
    printf("%f\n", area);
    return 0;
}

float circle(int r)
{
    float a;
    a = 3.14 * r * r;
    return (a);
}
```

Output:

3.140000

(d)

```
# include <stdio.h>
int main()
{
    void slogan();
    int c = 5;
    c = slogan();
    printf("%d\n", c);
    return 0;
}

void slogan()
{
    printf("Only He men use C!\n");
}
```

Output:

Error message by compiler

## Chapter - 09

### Pointers

What will be the output of the following programs?

(a)

```
#include <stdio.h>
void fun(int, int);

int main()
{
    int i = 5, j = 2;
    fun(i, j);
    printf("%d %d\n", i, j);
    return 0;
}

void fun(int i, int j)
{
    i = i * i;
    j = j * j;
}
```

Output:

5 2

(b)

```
#include <stdio.h>
void fun(int *, int *);

int main()
{
    int i = 5, j = 2;
    fun(&i, &j);
    printf("%d %d\n", i, j);
    return 0;
}

void fun(int *i, int *j)
{
    *i = *i * *i;
    *j = *j * *j;
}
```

Output:

25 4

(c)

```
#include <stdio.h>
int main()
{
    float a = 13.5;
    float *b, *c;
    b = &a; /* suppose address of a is 1006 */
    c = b;
    printf("%u %u %u\n", &a, b, c);
    printf("%f %f %f %f %f\n", a, *(&a), *&a, *b, *c);
    return 0;
}
```

*Output:*

```
1006 1006 1006
13.500000 13.500000 13.500000 13.500000 13.500000
```

Note : Instead of 1006 you may get some other number

## Chapter - 10

### Recursion

What will be the output of the following programs?

(a)

```
#include <stdio.h>
int main()
{
    printf("C to it that C survives\n");
    main();
    return 0;
}
```

*Output:*

The message will get printed indefinitely.

(b)

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int i = 0;
    i++;
    if (i <= 5)
    {
        printf("C adds wings to your thoughts\n");
        exit(0);
        main();
    }
    return 0;
}
```

*Output:*

C adds wings to your thoughts



## Chapter - 11

### Data Types Revisited

What will be the output of the following programs?

(a)

```
#include <stdio.h>
int main()
{
    int i;
    for (i = 0; i <= 50000; i++)
        printf("%d\n", i);
    return 0;
}
```

Output:

```
0
1
...
...
50000
```

(b)

```
#include <stdio.h>
int main()
{
    float a = 13.5;
    double b = 13.5;
    printf("%f %lf\n", a, b);
    return 0;
}
```

Output:

```
13.500000 13.500000
```

(c)

```
#include <stdio.h>
int i = 0;
void val();

int main()
{
    printf("main's i = %d\n", i);
    i++;
    val();
}
```

```

        printf("main's i = %d\n", i);
        val();
        return 0;
}

void val()
{
    i = 100;
    printf("val's i = %d\n", i);
    i++;
}

```

*Output:*

```

main's i = 0
val's i = 100
main's i = 101
val's i = 100

```

(d)

```

#include <stdio.h>

int f(int);
int g(int);

int main()
{
    int x, y, s = 2;
    s *= 3;
    y = f(s);
    x = g(s);
    printf("%d %d %d\n", s, y, x);
    return 0;
}

int t = 8;

int f(int a)
{
    a += -5;
    t -= 4;
    return (a + t);
}

int g(int a)
{
    a = 1;
    t += a;
    return (a + t);
}

```

Output:

6 5 6

(e)

```
#include <stdio.h>
int main()
{
    static int count = 5;
    printf("count = %d\n", count--);
    if (count != 0)
        main();
    return 0;
}
```

Output:

count = 5  
count = 4  
count = 3  
count = 2  
count = 1

(f)

```
#include <stdio.h>
int g(int);

int main()
{
    int i, j;
    for (i = 1; i < 5; i++)
    {
        j = g(i);
        printf("%d\n", j);
    }
    return 0;
}

int g(int x)
{
    static int v = 1;
    int b = 3;
    v += x;
    return (v + x + b);
}
```

Output:

6  
9  
13  
18

(g)

```
#include <stdio.h>
```

```
int main()
{
    func();
    func();
    return 0;
}
```

```
void func()
{
    auto int i = 0;
    register int j = 0;
    static int k = 0;
    i++;
    j++;
    k++;
    printf("%d %d %d\n", i, j, k);
}
```

Output:

1 1 1  
1 1 2

(h)

```
#include <stdio.h>
```

```
int x = 10;
```

```
int main()
{
    int x = 20;
    {
        int x = 30;
        printf("%d\n", x);
    }
    printf("%d\n", x);
    return 0;
}
```

*Output:*

30  
20

## Chapter - 12

### The C Preprocessor

What will be the output of the following programs?

(a)

```
#include <stdio.h>
int main()
{
    int i = 2;
    #ifdef DEF
        i *= i;
    #else
        printf("%d\n", i);
    #endif
    return 0;
}
```

Output:

2

(b)

```
#include <stdio.h>
#define PRODUCT(x) ( x * x )

int main()
{
    int i = 3, j, k, l;
    j = PRODUCT(i + 1);
    k = PRODUCT(i++);
    l = PRODUCT(++i);
    printf("%d %d %d %d\n", i, j, k, l);
    return 0;
}
```

Output:

7 7 9 49

or

7 7 12 49

(c)

```
#include <stdio.h>
#define PI 3.14
#define AREA( x, y, z ) ( PI * x * x + y * z ) ;

int main()
{
    float a = AREA(1, 5, 8);
    float b = AREA(AREA ( 1, 5, 8 ), 4, 5);
    printf(" a = %f\n", a);
    printf(" b = %f\n", b);
    return 0;
}
```

*Output:*

Error. Since there is a semicolon in the macro definition of **AREA**. If we drop the semicolon then the program will compile successfully. Nested macros are allowed.

## Chapter - 13

### Arrays

What will be the output of the following programs?

(a)

```
#include <stdio.h>

int main()
{
    int num[26], temp;
    num[0] = 100;
    num[25] = 200;
    temp = num[25];
    num[25] = num[0];
    num[0] = temp;
    printf("%d %d\n", num[0], num[25]);
    return 0;
}
```

Output:

200 100

(b)

```
#include <stdio.h>

int main()
{
    int array[26], i;
    for (i = 0; i <= 25; i++)
    {
        array[i] = 'A' + i;
        printf("%d %c\n", array[i], array[i]);
    }
    return 0;
}
```

Output:

65 A  
66 B  
...  
...  
90 Z



(c)

```
#include <stdio.h>
int main()
{
    int sub[50], i;
    for (i = 0; i <= 48; i++);
    {
        sub[i] = i;
        printf("%d\n", sub[i]);
    }
    return 0;
}
```

*Output:*

49

Since I takes a value 49 when it reaches the statement **sub[ i ] = i ;**

What will be the output of the following programs?

(a)

```
#include <stdio.h>
int main()
{
    int b[] = { 10, 20, 30, 40, 50 };
    int i;
    for (i = 0; i <= 4; i++)
        printf("%d\n", *(b + i));
    return 0;
}
```

*Output:*

10  
20  
30  
40  
50

(b)

```
#include <stdio.h>
int main()
{
    int b[] = { 0, 20, 0, 40, 5 };
    int i, *k;
    k = b;
```

```

    for (i = 0; i <= 4; i++)
    {
        printf("%d\n", *k);
        k++;
    }
    return 0;
}

```

Output:

```

0
20
0
40
5

```

(c)

```

#include <stdio.h>

void change(int *, int);

int main()
{
    int a[] = { 2, 4, 6, 8, 10 };
    int i;
    change(a, 5);
    for (i = 0; i <= 4; i++)
        printf("%d\n", a[i]);
    return 0;
}

void change(int *b, int n)
{
    int i;
    for (i = 0; i < n; i++)
        *(b + i) = *(b + i) + 5;
}

```

Output:

```

7
9
11
13
15

```

(d)

```
#include <stdio.h>
int main()
{
    static int a[5];
    int i;
    for (i = 0; i <= 4; i++)
        printf("%d\n", a[i]);
    return 0;
}
```

Output:

0  
0  
0  
0  
0

(e)

```
#include <stdio.h>
int main()
{
    int a[5] = { 5, 1, 15, 20, 25 };
    int i, j, k = 1, m;
    i = ++a[1];
    j = a[1]++;
    m = a[i++];
    printf("%d %d %d\n", i, j, m);
}
```

Output:

3 2 15

## Chapter - 14

### Multidimensional Arrays

What will be the output of the following programs?

(a)

```
#include <stdio.h>
int main()
{
    int n[3][3] = { 2, 4, 3, 6, 8, 5, 3, 5, 1 };
    printf("%d %d %d\n", *n, n[3][3], n[2][2]);
    return 0;
}
```

Output:

<base address><garbage value> 1

(b)

```
#include <stdio.h>
int main()
{
    int n[3][3] = { 2, 4, 3, 6, 8, 5, 3, 5, 1 };
    int i, *ptr;
    ptr = n;
    for (i = 0; i <= 8; i++)
        printf("%d\n", *(ptr + i));
    return 0;
}
```

Output:

2  
4  
3  
6  
8  
5  
3  
5  
1

(c)

```
#include <stdio.h>
int main() {
    int n[3][3] = { 2, 4, 3, 6, 8, 5, 3, 5, 1 };
    int i, j;
    for (i = 0; i <= 2; i++)
        for (j = 0; j <= 2; j++)
            printf("%d %d\n", n[i][j], (*(n + i) + j));
    return 0;
}
```

*Output:*

```
2 2
4 4
3 3
6 6
8 8
5 5
3 3
5 5
1 1
```

## Chapter - 15

### Strings

What will be the output of the following programs?

(a)

```
#include <stdio.h>
int main()
{
    char c[2] = "A";
    printf("%c\n", c[0]);
    printf("%s\n", c);
    return 0;
}
```

*Output:*

A  
A

(b)

```
#include <stdio.h>
int main()
{
    char s[] = "Get organised! learn C!!";
    printf("%s\n", &s[2]);
    printf("%s\n", s);
    printf("%s\n", &s);
    printf("%c\n", s[2]);
    return 0;
}
```

*Output:*

t organised! learn C!!  
Get organised! learn C!!  
Get organised! learn C!!  
t

```

(c)
#include <stdio.h>
int main()
{
    char s[] = "No two viruses work similarly";
    int i = 0;
    while (s[i] != 0)
    {
        printf("%c %c\n", s[i], *(s + i));
        printf("%c %c\n", i[s], *(i + s));
        i++;
    }
    return 0;
}

```

Output:

```

N N
N N
o o
o o
t t
t t
...
...
y y
y y

```

```

(d)
#include <stdio.h>
int main()
{
    char s[] = "Churchgate: no church no gate";
    char t[25];
    char *ss, *tt;
    ss = s;
    while (*ss != '\0')
        *tt++ = *ss++;
    printf("%s\n", t);
    return 0;
}

```

Output:

The code causes an exception as The variable 'tt' is being used without being initialized.

or

No output

(e)

```

#include <stdio.h>

```

```

int main()
{
    char str1[] = { 'H', 'e', 'l', 'l', 'o', 0 };
    char str2[] = "Hello";
    printf("%s\n", str1);
    printf("%s\n", str2);
    return 0;
}

```

Output:

Hello  
Hello

(f)

```

#include <stdio.h>

```

```

int main()
{
    printf(5 + "Good Morning ");
    return 0;
}

```

Output:

Morning

(g)

```

#include <stdio.h>
int main()
{
    printf("%c\n", "abcdefgh" [4]);
    return 0;
}

```

Output:

e

(h)

```

#include <stdio.h>
int main()
{
    printf("%d %d %d\n", sizeof('3'), sizeof("3"), sizeof(3));
    return 0;
}

```

Output:



2 2 2

or

4 2 4

## **Chapter - 16**

### **Handling Multiple Strings**

## Chapter - 17

### Structures

What will be the output of the following programs?

(a)

```
#include <stdio.h>
#include <string.h>
int main()
{
    struct gospel
    {
        int num;
        char mess1[50];
        char mess2[50];
    } m;
    m.num = 1;
    strcpy(m.mess1, "If all that you have is hammer");
    strcpy(m.mess2, "Everything looks like a nail");
    /* assume that the structure is located at address 1004 */
    printf("%u %u %u\n", &m.num, m.mess1, m.mess2);
    return 0;
}
```

*Output:*

Address of each structure element would be printed out

(b)

```
#include <stdio.h>
#include <string.h>

int main()
{
    struct part
    {
        char partname[50];
        int partnumber;
    };
    struct part p, *ptrp;
    ptrp = &p;
    strcpy(p.partname, "CrankShaft");
    p.partnumber = 102133;
    printf("%s %d\n", p.partname, p.partnumber);
    printf("%s %d\n", (*ptrp).partname, (*ptrp).partnumber);
    printf("%s %d\n", ptrp->partname, ptrp->partnumber);
    return 0;
}
```

*Output:*

```
CrankShaft 102133  
CrankShaft 102133  
CrankShaft 102133
```

(c)

```
#include <stdio.h>  
  
struct gospel  
{  
    int num;  
    char mess1[50];  
    char mess2[50];  
} m1 = { 2, "If you are driven by success", "make sure that it is a  
quality drive" };  
  
int main()  
{  
    struct gospel m2, m3;  
    m2 = m1;  
    m3 = m2;  
    printf("%d %s %s\n", m1.num, m2.mess1, m3.mess2);  
    return 0;  
}
```

*Output:*

```
2 If you are driven by success make sure that it is a quality drive
```

## Chapter - 18

### Console Input/Output

What will be the output of the following programs?

(a)

```
#include <stdio.h>
#include <ctype.h>

int main()
{
    char ch;
    ch = getchar();
    if (islower(ch))
        putchar(toupper(ch));
    else
        putchar(tolower(ch));
    return 0;
}
```

Output:

a  
A  
Z  
z

(b)

```
#include <stdio.h>
int main()
{
    int i = 2;
    float f = 2.5367;
    char str[] = "Life is like that";
    printf("%d\t%3.3f\t%4s\n", i, f, str);
    return 0;
}
```

Output:

2      2.537      Life is like that

(c)

```
#include <stdio.h>

int main()
{
    printf ( "More often than \b\b not \rthe person who \
            wins is the one who thinks he can!\n" ) ;
    return 0;
}
```

*Output:*

the person who wins is the one who thinks he can!

(d)

```
#include <stdio.h>
#include <conio.h>
char p[] = "The sixth sick sheikh's sixth ship is sick";
int main()
{
    int i = 0;
    while (p[i] != '\0')
    {
        putchar(p[i]);
        i++;
    }
    return 0;
}
```

*Output:*

The sixth sick sheikh's sixth ship is sick

**Chapter - 19**  
**File Input/Output**

## **Chapter - 20**

### **More Issues in Input/Output**



**Chapter - 21**  
**Operations On Bits**

## Chapter - 22

### Miscellaneous Features

What will be the output of the following programs?

(a)

```
#include <stdio.h>
int main()
{
    enum status
    {
        pass, fail, atkt
    };
    enum status stud1, stud2, stud3;
    stud1 = pass;
    stud2 = fail;
    stud3 = atkt;
    printf("%d %d %d\n", stud1, stud2, stud3);
    return 0;
}
```

Output:

0 1 2

(b)

```
#include <stdio.h>
int main()
{
    printf("%f\n", (float) ((int) 3.5 / 2));
    return 0;
}
```

Output:

1.000000

(c)

```
#include <stdio.h>
int main()
{
    float i, j;
    i = (float) 3 / 2;
    j = i * 3;
    printf("%d\n", (int) j);
    return 0;
}
```

*Output:*

4

## Chapter - 23

### C Under Linux

**Chapter - 24**  
**Periodic Tests**





# RAJALAKSHMI

## ENGINEERING COLLEGE

*Website : [www.rajalakshmi.org](http://www.rajalakshmi.org)*

*Elearning : [www.rajalakshmiengg.com](http://www.rajalakshmiengg.com)*

***Give a man a fish and you feed him for a day.  
Teach him how to fish and you feed him for a lifetime.***