

- (d) The * operator lets us access the value present at an address in memory with an intension of reading it or modifying it.
- (e) A function can be called either by value or by reference.
- (f) Pointers can be used to make a function return more than one value simultaneously in an indirect manner.

Exercise

[A] 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;
 }

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;
 }

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;
}

```

1006 1006 1006

[B] Point out the errors, if any, in the following programs:

(a) #include <stdio.h>
void pass (int, int);
 int main()

pass() can not return a value

```

{
    int i = 135, a = 135, k;
    k = pass ( i, a );
    printf ( "%d\n", k );
    return 0;
}

```

```

void pass ( int j, int b )
int c;
{
    c = j + b;
    return ( c );
}

```

(b) #include <stdio.h>
 void jlaayjo (int, int)
 int main()

```

{
    int p = 23, f = 24;
    jlaayjo ( &p, &f );
    printf ( "%d %d\n", p, f );
    return 0;
}

```

can't pass address in the function

```

void jlaayjo ( int q, int g )
{
    q = q + q;
    g = g + g;
}

```

```

(c) #include <stdio.h>
void check ( int ) ;
int main( )
{
    int k = 35, z ;
    z = check ( k ) ;
    printf ( "%d\n", z ) ;
    return 0 ;
}

```

Can't assign
function

```

void check ( m )
{
    int m ;
    if ( m > 40 )
        return ( 1 ) ;
    else
        return ( 0 ) ;
}

```

```

(d) #include <stdio.h>
void function ( int * ) ;
int main( )
{
    int i = 35, *z ;
    z = function ( &i ) ;
    printf ( "%d\n", z ) ;
    return 0 ;
}
void function ( int *m )
{
    return ( *m + 2 ) ;
}

```

Can't assign function

[C] Attempt the following:

- Write a function that receives 5 integers and returns the sum, average and standard deviation of these numbers. Call this function from `main()` and print the results in `main()`.
- Write a function that receives marks received by a student in 3 subjects and returns the average and percentage of these marks. Call this function from `main()` and print the results in `main()`.
- Write a C function to evaluate the series

$$\sin(x) = x - (x^3 / 3!) + (x^5 / 5!) - (x^7 / 7!) + \dots$$

up to 10 terms.

- (d) Given three variables x , y , z write a function to circularly shift their values to right. In other words if $x = 5$, $y = 8$, $z = 10$, after circular shift $y = 5$, $z = 8$, $x = 10$. Call the function with variables a , b , c to circularly shift values.
- (e) If the lengths of the sides of a triangle are denoted by a , b , and c , then area of triangle is given by

$$area = \sqrt{S(S-a)(S-b)(S-c)}$$

where, $S = (a + b + c) / 2$. Write a function to calculate the area of the triangle.

- (f) Write a function to compute the distance between two points and use it to develop another function that will compute the area of the triangle whose vertices are $A(x_1, y_1)$, $B(x_2, y_2)$, and $C(x_3, y_3)$. Use these functions to develop a function which returns a value 1 if the point (x, y) lies inside the triangle ABC, otherwise returns a value 0.
- (g) Write a function to compute the greatest common divisor given by Euclid's algorithm, exemplified for $J = 1980$, $K = 1617$ as follows:

$1980 / 1617 = 1$	$1980 - 1 * 1617 = 363$
$1617 / 363 = 4$	$1617 - 4 * 363 = 165$
$363 / 165 = 2$	$363 - 2 * 165 = 33$
$5 / 33 = 5$	$165 - 5 * 33 = 0$

Thus, the greatest common divisor is 33.