**Tutorial 2 – Functions and Pointers**

1. Assume the following declaration:

int number;

int \*p;

Assume also that the address of number is 7700 and the address of p is 3478. That is,

|  |
| --- |
|  |

3478 p

.

.

.

|  |
| --- |
|  |

7700 number

For each case below, determine the value of

(a) number (b) &number (c) p (d) &p (e) \*p

All of the results are cumulative.

* 1. p = 100; number = 8

1. 8
2. 7700
3. 100
4. 3478
   1. number = p
5. 100
6. 7700
7. 100
8. 3478
   1. p = &number
9. 100
10. 7700
11. 7700
12. 3478
13. 100
    1. \*p = 10
14. 10
15. 7700
16. 7700
17. 3478
18. 10
    1. number = &p
19. 3478
20. 7700
21. 7700
22. 3478
23. 3478
    1. p = &p
24. 3478
25. 7700
26. 3478
27. 3478
28. 3478

1. Find the error in each of the following program segments and explain how the error may be corrected.

* 1. int product(int m, int n)

{

int result;

result =m \* n;

return result;

}

* 1. int sumofSquare(int n) /\* assume n is non‐negative \*/

{

int sum = 0, j;

if (n == 0)

return 0;

else

for (j = 1; j <= n; j++) sum += j \* j ;

return sum;

}

* 1. void ft(float a)

{

~~float a;~~

printf(“%f\n”, a);

}

* 1. void height(float \* h)

{

scanf(“%f”, ~~&~~h);

}

* 1. void height(float \* h)

{

scanf(“%f”, h);

~~return \*h;~~

}

* 1. int divideBy4(int n)

{

~~int divideBy2(int m)~~

~~{~~

~~return m/2;~~

~~}~~

Function body is not allowed inside another function

return (divideBy2(divideBy2(n));

}

1. What will be the output of the following program?

|  |
| --- |
| #include <stdio.h> void function0(); void function1(int h, int k); void function2(int \*h, int \*k); int main()  {  int h, k;    h = 5;  k = 15;  printf(“h = %d, k = %d\n”, h, k); /\* line (i) \*/  function0();  printf(“h = %d, k = %d\n”, h, k); /\* line (ii) \*/  function1(h, k);  printf(“h = %d, k = %d\n”, h, k); /\* line (iii) \*/ function2(&h, &k);  printf(“h = %d, k = %d\n”, h, k); /\* line (iv) \*/  return 0;  }  void function0()  {  int h, k; |

h = k = ‐100;

printf(“h = %d, k = %d\n”, h, k); /\* line (v) \*/

}

void function1(int h, int k)

{

printf(“h = %d, k = %d\n”, h, k); /\* line (vi) \*/

h = k = 100;

printf(“h = %d, k = %d\n”, h, k); /\* line (vii) \*/

}

void function2(int \*h, int \*k)

{

printf(“h = %d, k = %d\n”, \*h, \*k); /\* line (viii) \*/

\*h = \*k = 200;

printf(“h = %d, k = %d\n”, \*h, \*k); /\* line (ix) \*/

}

i) h = 5, k = 15

v) h = -100, k = -100

ii) h = 5, k = 15

vi) h = 5, k = 15

vii) h = 100, k = 100

iii) h = 5, k = 15

viii) h = 5, k = 15

ix) h = 200, k = 200

iv) h = 200, k = 200

1. **(calDistance)** Write a C program that accepts four decimal values representing the coordinates of two points, i.e. (x1, y1) and (x2, y2), on a plane, and calculates and displays the distance between the points:

distance = (x2 -x1)2 + (y2 - y1)2

Your program should be implemented using functions. Provide two versions of the function for calculating the distance: (a) one uses call by value only for passing parameters; and (b) the other uses call by reference to pass the result to the calling function.