

COM2002 INTERMEDIATE PROGRAMMING
2024 – 2025 SPRING
C PROGRAMMING EXERCISE - 01

Topic : Pointers

Exercise-1 : Write a statement to declare a variable `i` as integer and a pointer `p` points to `i`.

Exercise-2 : If `i` is an `int` variable and `p` points to `i`, which of the following expressions are aliases for `i`?

- a. `int value = *p;`
- b. `float value = *p;`
- c. `int value = p;`
- d. `float value = p;`
- e. `int value = *&i;`
- f. `float value = *&i;`
- g. `int value = &p;`

Exercise-3 : What is the output of the code fragment?

```
int value = 6, *p = &value, *q;  
printf("%d\n", p);  
printf("%d\n", *p);  
printf("%d\n", &p);  
printf("%d\n", value);  
printf("%d\n", &value);  
printf("%d\n", *value);  
printf("%d\n", &*value);  
printf("%d\n", q);  
printf("%d\n", *q);  
printf("%d\n", &q);
```

Exercise-4 : The function determines the smallest number of \$20, \$10, \$5, and \$1 bills necessary to pay the amount represented by the *dollars* parameter. The *twenties* parameter points to a variable in which the function will store the number of \$20 bills required. The *tens*, *fives*, and *ones* parameters are similar. The function contains several errors, find and correct them.

```
void pay_amount(int dollars, int *twenties, int *tens, int *fives, int  
*ones){  
    twenties = dollars / 20;  
    dollars -= twenties * 20;  
  
    tens = dollars / 10;  
    dollars -= tens * 10;  
  
    fives = dollars / 5;  
    ones = dollars % 5;  
}
```

Exercise-5 : The prototype of the `find_largest` is

```
int* find_largest(int a[], int n);
```

Call the `find_largest` function to find the maximum value of the given array

```
int values[] = { 0, 3, 28, 1, 30, 41, 20, 14, 301 };
```

Exercise-6 : What is the output of the following fragment?

```
int m = 10, n = 5;
int* mp, * np;
mp = &m;
np = &n;
*mp = *mp + *np;
*np = *mp - * np;
printf("%d %d\n%d %d\n", m, *mp, n, *np);
```

Exercise-7 : Given the declarations

```
int m = 25, n = 77;
int* itemp;
```

describe the errors in each of the following statements

```
m = &n;
itemp = m;
*itemp = *&m;
*itemp = &n;
```

Exercise-8 : What is the output of the program?

```
#include <stdio.h>

void function2(int* ptr, int y);
void function(int* x, int* y);
int main(void)
{
    int x, y;
    function(&x, &y);
    printf("x = %d, y = %d\n", x, y);
    return (0);
}

void function2(int* ptr, int y)
{
    int x;
    x = 10;
    *ptr = 2 * x - y;
}

void function(int* x, int* y)
{
    function2(x, 7);
    function2(y, *x);
}
```

Exercise-9 : The definition of the `find_middle` is

```
int* find_middle(int a[], int n) {  
    return &a[n / 2];  
}
```

Call the `find_middle` function to find the middle element of the given array

```
int values[] = { 0, 3, 28, 1, 30, 41, 20, 14, 301 };
```

Exercise-10 : Write a statement to declare an array `arr` with 10 elements as `float` and a pointer `p` points to the element 5 of the `arr` array.

DO NOT COPY