Programming In C – Midterm Preparation

Dated: 25-04-17

Q-1:

```
1. int main()
2. {
3.     print("Hello, world!");
4.     return 0;
5. }
6.
```

Q2:

```
 1. #include <stdio.h>
 2.
 3. int main() {
 4.     int a = 5, b = 8, c = 3, max;
 5.
 6.     if (a > b)
 7.         max = a;
 8.     else if (b > c)
 9.         max = b;
10.     else
11.         max = c;
12.
13.     printf("Maximum value is: %d\n", max);
14.     return 0;
15. }
16.
```

Q3:

```c
int main()
{
    int x = 10, y = 20;
    int result = add(x, y);
    printf("Result = %d\n", result);
    return 0;
}

int add(int a, int b) {
    return a + b;
}

```

Q4:

```c
#include <stdio.h>

int main() {
    int i = 1, sum = 0;

    while (i <= 10)
        sum = i + 1;
        i++;

    printf("Sum from 1 to 10 is: %d\n", sum);
    return 0;
}

```

Q-5:

```c
#include <stdio.h>

int main() {
    int number = 10;
    float result;

    result = number / 4;
    printf("Result: %f\n", result);
    return 0;
}

```

Q-6:

```c
#include <stdio.h>

int main() {
    int a = 5;
    float b = 2.5;
    int result = a * b;

    printf("Result: %d\n", result);
    return 0;
}

```

Q-7:

```c
#include <stdio.h>

int main(void)
{
    printf("2/3 = %d\n", 2 / 3);
    return 0;
}

```

Q-8:

```c
#include <stdio.h>

int main(void) {
    int x, y, sum;
    sum = x + y;
    printf("합: %d\n", sum);
    return 0;
}

```

Q-9:

```c
#include <stdio.h>

int main(void) {
    int 1st_value = 10;
    int second value = 20;
    int sum = 1st_value + second value;
    printf("합: %d\n", sum);
    return 0;
}

```

Q-10:

```c
#include <stdio.h>

int main(void) {
    short a = 32767;
    a = a + 1;
    printf("a = %d\n", a);
    return 0;
}

```

Q-11:

```c
#include <stdio.h>

int main(void) {
    double f_temp = 90;
    double c_temp = 5 / 9 * (f_temp - 32);
    printf("섭씨온도: %f\n", c_temp);
    return 0;
}

```

Q-12:

```c
#include <stdio.h>

int main(void) {
    int x = 5;
    if (2 < x < 10)
        printf("x is between 2 and 10\n");
    else
        printf("x is NOT between 2 and 10\n");
    return 0;
}

```

Q-13:

```c
#include <stdio.h>

int main(void) {
    int x = 5, y = 5;
    if (x = y)
        printf("Equal\n");
    else
        printf("Not equal\n");
    return 0;
}

```

Q-14:

```c
#include <stdio.h>

int main(void) {
    int a = 2, b = 3;
    int result = a + b * 2 == 8;
    printf("result = %d\n", result);
    return 0;
}

```

Q-15:

```c
#include <stdio.h>

int main(void) {
    float x = 10.5;
    int result = x % 3;
    printf("Result = %d\n", result);
    return 0;
}

```

Q-16:

```c
#include <stdio.h>

int main(void) {
    int x = 5;
    if (x > 10 && ++x < 20)
        printf("Inside if: x = %d\n", x);
    else
        printf("Else block: x = %d\n", x);
    return 0;
}

```

Q-17:

```c
#include <stdio.h>

int main(void) {
    char data = 'A';
    char encrypted = data ^ 255;
    printf("Encrypted: %c\n", encrypted);
    char decrypted = encrypted ^ 254;  // Wrong key
    printf("Decrypted: %c\n", decrypted);
    return 0;
}

```

Q-18:

```c
#include <stdio.h>

int main(void) {
    int score = 85;

    if (score >= 80)
        if (score >= 90)
            printf("Grade: A\n");
        else
            printf("Grade: B\n");

    return 0;
}

```

Q-19:

```c
#include <stdio.h>

int main(void) {
    int a = 5, b = 0, result;

    if (b == 0);
        printf("Cannot divide by zero!\n");
    result = a / b;

    printf("Result: %d\n", result);
    return 0;
}

```

Q-20:

```c
#include <stdio.h>

int main(void) {
    int x = 5;

    if (2 < x < 10)
        printf("x is between 2 and 10\n");
    else
        printf("x is NOT between 2 and 10\n");

    return 0;
}

```

Q-21:

```c
#include <stdio.h>

int main(void) {
    int i = 1;
    while(i < 5) {
        printf("i = %d\n", i);
        // i++; // This line is missing
    }
    return 0;
}

```

Q-22:

```c
#include <stdio.h>

int main(void) {
    int n, i = 1;
    long fact = 1;

    printf("Enter an integer: ");
    scanf("%d", &n);

    while(i <= n); { // <-- Semicolon issue
        fact *= i;
        i++;
    }

    printf("%d! = %ld\n", n, fact);
    return 0;
}

```

Q-23:

```c
#include <stdio.h>

int main(void) {
    printf("C to F: %lf\n", c_to_f(36.0));
    return 0;
}

double c_to_f(double c) {
    return 9.0 / 5.0 * c + 32;
}

```

Q-24:

```c
#include <stdio.h>

int combination(int n, int r) {
    return factorial(n) / (factorial(n - r) * factorial(r));
}

int main(void) {
    int n = 5, r = 2;
    printf("C(%d, %d) = %d\n", n, r, combination(n, r));
    return 0;
}

```

Q-25:

```
1. char choice;
2. printf("Select menu: ");
3. choice = getchar();
4. if (choice == 'q') break;
```

Q-26:

```
 1. #include <stdio.h>
 2.
 3. int max(int a, int b) {
 4.     if (a > b)
 5.         return a;
 6.     else
 7.         printf("%d", b);
 8. }
 9.
10. int main(void) {
11.     int result = max(3, 9);
12.     printf("Max = %d\n", result);
13.     return 0;
14. }
15.
```

Q-27:

```
1. #include <stdio.h>
2.
3. int counter;
4.
5. int main(void) {
6.     printf("counter = %d\n", counter);
7.     return 0;
8. }
9.
```

Q-28;

```
 1. #include <stdio.h>
 2.
 3. void increase(int i) {
 4.     i++;
 5. }
 6.
 7. int main(void) {
 8.     int i = 5;
 9.     increase(i);
10.     printf("i = %d\n", i);
11.     return 0;
12. }
13.
```

Q-29:

```c
#include <stdio.h>

void visit() {
    int count = 0;
    count++;
    printf("Visit count: %d\n", count);
}

int main(void) {
    visit();
    visit();
    visit();
    return 0;
}

```

Q-30:

```c
#include <stdio.h>

int value = 100;

int main(void) {
    int value = 42;
    printf("value = %d\n", value);
    return 0;
}

```

Q-31:

```c
#include <stdio.h>

long factorial(int n);

int main(void) {
    printf("5! = %ld\n", factorial(5));
    return 0;
}

void factorial(int n) {
    if (n <= 1)
        return 1;
    else
        return n * factorial(n - 1);
}

```

Q-32:

```c
#include <stdio.h>

void save(int amount) {
    long balance = 0;
    balance += amount;
    printf("Balance: %ld\n", balance);
}

int main(void) {
    save(1000);
    save(2000);
    save(3000);
    return 0; }
```

Q-33:

```c
// in file1.c
int count = 0;

// in file2.c
#include <stdio.h>

int main(void) {
    extern int count;
    count = 5;
    printf("count = %d\n", count);
    return 0;
}

```

Q-34:

What's the output of this recursive call?

```c
#include <stdio.h>

void print_binary(int x) {
    if (x > 0) {
        print_binary(x / 2);
        printf("%d", x % 2);
    }
}

int main(void) {
    print_binary(5);
    return 0;
}
```

Q-35: This code tries to store a value in an array. What's the issue?

```c
#include <stdio.h>

int main(void) {
    int scores[5];
    scores[5] = 100;
    return 0;
}

```

Q-36: This array should be filled with random values, but the output is weird. Why?

```c
#include <stdio.h>
#define SIZE 5

int main(void) {
    int scores[SIZE];
    int i;

    for (i = 0; i < SIZE; i++)
        printf("scores[%d] = %d\n", i,
scores[i]);

    return 0;
}

```

Q-37: This code should copy one array to another. What's the error?

```c
#include <stdio.h>
#define SIZE 5

int main(void) {
    int a[SIZE] = {1, 2, 3, 4, 5};
    int b[SIZE];

    b = a;

    return 0;
}

```

Q-38: This function compares two arrays. What's wrong?

```c
#include <stdio.h>
#define SIZE 5

int main(void) {
    int a[SIZE] = {1, 2, 3, 4, 5};
    int b[SIZE] = {1, 2, 3, 4, 5};

    if (a == b)
        printf("Arrays are equal.\n");
    else
        printf("Arrays are not equal.\n");

    return 0;
}

```

Q-39: This selection sort implementation gives a segmentation fault. Why?

```
 1. #include <stdio.h>
 2. #define SIZE 10
 3.
 4. int main(void) {
 5.     int list[SIZE] = {3, 2, 9, 7, 1, 4, 8, 0, 6,
5};
 6.     int i, j, temp, least;
 7.
 8.     for (i = 0; i <= SIZE - 1; i++) {
 9.         least = i;
10.         for (j = i + 1; j < SIZE; j++)
11.             if (list[j] < list[least])
12.                 least = j;
13.
14.         temp = list[i];
15.         list[i] = list[least];
16.         list[least] = temp;
17.     }
18.
19.     return 0;
20. }
21.
```

Q-40: What's wrong with this 2D array?

```c
#include <stdio.h>

int main(void) {
    int matrix[3][3] = {
        {1, 2},
        {4, 5, 6},
        {7}
    };

    printf("%d\n", matrix[1][2]);
    return 0;
}

```

Q-41: This function tries to find the max in an array. What's wrong?

```c
#include <stdio.h>
#define SIZE 5

int max(int arr[]) {
    int i, maximum = arr[0];
    for (i = 1; i < SIZE; i++) {
        if (arr[i] > maximum)
            maximum = arr[i];
    }
    return maximum;
}

int main(void) {
    int numbers[SIZE] = {2, 5, 3, 8, 6};
    printf("Max = %d\n", max(numbers));
    return 0;
}

```

Q-42: What's wrong with this binary search implementation?

```c
#include <stdio.h>
#define SIZE 10

int binary_search(int arr[], int key) {
    int low = 0, high = SIZE - 1, mid;
    while (low <= high) {
        mid = (low + high) / 2;
        if (arr[mid] == key)
            return mid;
        else if (arr[mid] < key)
            low = mid + 1;
        else
            high = mid - 1;
    }
    return -1;
}

int main(void) {
    int data[SIZE] = {1, 2, 3, 4, 5, 6, 7, 8, 9};
// only 9 elements!
    int result = binary_search(data, 6);
    printf("Result: %d\n", result);
    return 0;
}

```