



Attempt 5 of the following questions (including questions 1 and 2)

No of Questions: 6 in 6 page(s)
Total Mark: 60

Question 1:

(12 Marks)

Determine the output for each of the following code snippets (assuming successful compilation):

a) (2 Marks)

```
for (int i = 0; i < 5; i++) {
    for (int j = 0; j <= 5; j++) {
        if (j == i + 1) break;
        cout << j;
    }
    cout << endl;
}
```

b) (2 Marks)

```
for (int i = 0; i < 5; i++) {
    for (int j = 0; j <= 5; j++) {
        if (j == i) continue;
        cout << j;
    }
    cout << endl;
}
```

c) (2 Marks)

```
int i = 0;
do {
    cout << ++i << endl;
    cout << i++ << endl;
} while (i < 5);
```

d) (2 Marks)

```
for (int i = 0; i < 5; i++) {
    for (int j = 0; j < 5; j++)
        cout << ((i+j) % 2 == 0 ? "■" : " ");
    cout << endl;
}
```

e) (2 Marks)

```
for (int i = 0; i < 5; i++) {
    for (int j = 0; j < 5; j++) {
        char c;
        switch (abs(i - j)) {
            case 0 : c = '\\'; break;
            case 2 : c = '+' ; break;
            case 4 : c = '.' ; break;
            default: c = ' ' ; break;
        }
        cout << c;
    }
    cout << endl;
}
```

f) (2 Marks)

```
{
    int f1 = 1;
    int f2 = 1;
    for (int i = 1; i <= 5; i += 1) {
        cout << f1 << endl;
        f2 = f1 + f2;
        f1 = f2 - f1;
    }
}
```

Solution:

a)

```
0
01
012
0123
01234
```

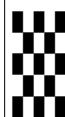
b)

```
12345
02345
01345
01245
01235
```

c)

```
1
1
3
3
5
5
```

d)



e)

```
\ + .
 \ +
+ \ +
+ \
. + \
```

f)

```
1
1
2
3
5
```

Question 2:

(12 Marks)

For the three questions that you will solve later:

- a) avoid syntax and runtime errors,
- b) validate the user input,
- c) prompt the user with meaningful instructions, and
- d) write the code using a clean style.

(4 Marks)

(4 Marks)

(4 Marks)

(4 Marks)^{\$}

Solution:

For example, the student should

- a)
 - include necessary libraries,
 - use the appropriate namespace,
 - provide function prototype whenever needed,
 - declare any used variables,
 - write every statement in the right syntax,
- b) check that every value entered by the user falls within the acceptable range,
- c)
 - print a meaningful hint before reading any input from the user,
 - print a meaningful label before printing any output to the user,
 - print a meaningful message whenever something wrong happens,
- d)
 - use indentation,
 - follow naming conventions,
 - align braces, and
 - write comments.

Question 3:

(12 Marks)

Write a full program including three functions for printing the following patterns using only one '*' and one ' ' per function.

a) (4 Marks)

```
*****
*   *
* * *
*   *
*****
```

b) (4 Marks)

```
*   *
*   *
*
*   *
*   *
```

c) (4 Marks)

```
*
***
*****
***
*
```

Solution:

```
#include <iostream>
#include <cmath>

using namespace std;

//Function Prototype
void q_3_a();
void q_3_b();
void q_3_c();

int main() {
    cout << "a)\n";
    q_3_a();
    cout << "b)\n";
    q_3_b();
    cout << "c)\n";
    q_3_c();
    return 0;
}

//Function Definition
//a)
void q_3_a() {
    for (int i = 0; i < 5; i++) {
        for (int j = 0; j < 5; j++)
            cout << ((i == 0 || i == 4 || j == 0 || j == 4 || (i == 2 && j == 2)) ? '*' : ' ');
        cout << endl;
    }
}

//b)
void q_3_b() {
    for (int i = 0; i < 5; i++) {
        for (int j = 0; j < 5; j++)
            cout << ((i - j == 0 || i + j == 4) ? '*' : ' ');
        cout << endl;
    }
}

//c)
void q_3_c() {
    for (int i = 0; i < 5; i++) {
        for (int j = 0; j < 5; j++)
            cout << ((abs(j - 2) + abs(i - 2) <= 2) ? '*' : ' ');
        cout << endl;
    }
}
```

Question 4:

(12 Marks)

a)

(6 Marks)

Write a function `weekday` that takes an integer $d \in [1, 7]$ and returns a string representing the Arabic name.

b)

(6 Marks)

Write a full program that:

1. reads a weekday number from the user,
2. prints the weekday name returned from `weekday`, and
3. repeats if the user still needs to use the program.

Example: When the user enters 1, the program should print "السبت" (without the quotation marks)

Solution:

```
//b)
#include <iostream>

using namespace std;

//Function Prototype
char * weekday(int);

int main() {
    int d;
    char c;
    do {
        //Read weekday number
        //b)1.
        do {
            cout << "Enter weekday number [1,7]: ";
            cin >> d;
            if (d < 1) {
                cout << "Too small!" << endl;
            }
            if (d > 7) {
                cout << "Too large!" << endl;
            }
        } while (d < 1 || d > 7);

        //b)2.
        cout << "Weekday number " << d << " is: " << weekday(d) << endl;
        //Check if the user still needs to use the program
        //b)3.
        cout << "Enter Y|y to continue or anything else to end? ";
        cin >> c;
    } while (c == 'Y' || c == 'y');
    return 0;
}

//Function Definition
//a)
char * weekday(int d) {
    switch (d) {
        case 1:
            return "السبت";
        case 2:
            return "الأحد";
        case 3:
            return "الاثنين";
        case 4:
            return "الثلاثاء";
        case 5:
            return "الأربعاء";
        case 6:
            return "الخميس";
        case 7:
            return "الجمعة";
        default:
            return "Invalid!";
    }
}
```

Question 5:

(12 Marks)

Write a full program that reads an array of n numbers and prints its range. The range of an array is the difference between its maximum and minimum values. The program should consist of three functions:

- a) read for reading the array, (4 Marks)
- b) min for finding the minimum value, and (4 Marks)
- c) max for finding the maximum value. (4 Marks)

Solution:

```
#include <iostream>

using namespace std;

//Function Prototype
void read(float[], int);
float max(float[], int);
float min(float[], int);

int main() {
    const int N = 100;
    int n;
    float x[N];
    //Read array size
    do {
        cout << "Enter n [1," << N << "]: ";
        cin >> n;
        if (n < 1) {
            cout << "Too small!" << endl;
        }
        if (n > N) {
            cout << "Too large!" << endl;
        }
    } while (n < 1 || n > N);

    read(x, n);
    cout << "The range is: " << max(x, n) - min(x, n) << endl;
    return 0;
}

//Function Definition
//a)
void read(float x[], int n) {
    for (int i = 0; i < n; i++) {
        cout << "Enter element number " << i << ": ";
        cin >> x[i];
    }
}
//b)
float min(float x[], int n) {
    float minX = x[0];
    for (int i = 1; i < n; i++)
        if (minX > x[i])
            minX = x[i];
    return minX;
}
//c)
float max(float x[], int n) {
    float maxX = x[0];
    for (int i = 1; i < n; i++)
        if (maxX < x[i])
            maxX = x[i];
    return maxX;
}
```

Question 6:

(12 Marks)

In survey engineering, a *traverse* is an n -sided closed polygon. *Traverse angle balancing* is a process intended for adjusting (correcting) the measured internal angles of a given traverse according to the following equations. The target of this process is to make the actual sum of the corrected angles the same as the theoretical sum ($tsum$).

$$tsum = 180 \cdot (n - 2)$$

$$asum = \left(\sum_{i=1}^n a_i \right)$$

$$error = asum - tsum$$

$$correction = error/n$$

$$\hat{a}_i = a_i - correction \forall i \in [1, n]$$

Write a full program including three functions:

- a) (4 Marks)
read for reading three or more traverse angles a_1, a_2, \dots, a_n from the user,
- b) (4 Marks)
correct for performing *traverse angle balancing*, and
- c) (4 Marks)
print for printing the corrected angles $\hat{a}_1, \hat{a}_2, \dots, \hat{a}_n$ to the standard output.

Example: If the measured angles are $a = \{61.5, 60.5, 59.5\}$, then the corrected angles should be $\hat{a} = \{61.0, 60.0, 59.0\}$

Solution:

```
#include <iostream>

using namespace std;

//Function Prototype
void read(double [], int);
void correct(double [], int, double);
void print(double [], int);

int main() {
    const int N = 100;
    double a[N];
    int n, i;
    double tsum, sum = 0, error, correction;
    //Read traverse size
    do {
        cout << "How many angles [3," << N << "]: ";
        cin >> n;
        if (n < 3) {
            cout << "Too few angles!" << endl;
        }
        if (n > N) {
            cout << "Too many angles!" << endl;
        }
    } while (n < 3 || n > N);

    read(a, n);
    //Calculate actual and theoretical sums
    for (int i = 0; i <= n - 1; i += 1) {
        sum = sum + a[i];
    }
    tsum = 180 * (n - 2);
    //Calculate error and correction
    error = sum - tsum;
    correction = error / n;
    //Correct angles and print results
    correct(a, n, correction);
    cout << "Error = " << error << endl;
    cout << "Correction = " << correction << endl;
    cout << "Corrected angles:" << endl;
    print(a, n);
}
```

```
//Function Definition
//a)
void read(double a[], int n) {
    cout << "Enter " << n << " angles:" << endl;
    for (int i = 0; i <= n - 1; i += 1) {
        do {
            cout << "Angle " << i + 1 << " (0,360): ";
            cin >> a[i];
            if (a[i] <= 0 || a[i] >= 360) {
                cout << "Out of range!" << endl;
            }
        } while (a[i] <= 0 || a[i] >= 360);
    }
}
//b)
void correct(double a[], int n, double correction) {
    for (int i = 0; i <= n - 1; i += 1) {
        a[i] = a[i] - correction;
    }
}
//c)
void print(double a[], int n) {
    for (int i = 0; i <= n - 1; i += 1) {
        cout << a[i] << endl;
    }
}
```

Good Luck
Dr. Islam ElShaarawy