



United International University (UIU)

Dept. of Computer Science & Engineering (CSE)

Mid-term Exam : Trimester: Summer 2023

Course Code: CSE 1111, Course Title: STRUCTURED PROGRAMMING LANGUAGE

Time: 1 hour 45 min Total Marks: 30

Answer all the questions.

“Any examinee found adopting unfair means will be expelled from the trimester / program as per UIU disciplinary rules.”

1. (a) Which of the following are **invalid** variable names and **why**? [1]

(i) is-Val (ii) a1234 (iii) while (iv) _1num_new (v) CSE 1111

- (b) **Compute** the values of the variables **a, b, c, d** and **result**. [2]

(i) float a = 22/4;
(ii) int b = 2%7;
(iii) int e = 4, c = 11 + ++e;
(iv) int d = 2==3? 7:9;
(v) double result = 3!=5;

- (c) **Find output** of the following program for (i) **a = 0, b = 0**, and (ii) **a = -1, b = -7**: [3]

```
#include<stdio.h>
void main(){
    int a, b;
    scanf("%d%d", &a, &b);
    if(!(a-b) && ++a)
        printf("Pattern\n");
    if((a>0&&b>0)|| (a<0&&b<0)){
        printf("Fizz\n");
        if(a>0)
            printf("Positive\n");
        return 0;
        if(b<0)
            printf("Negative\n");
    }
    else if(a>0 && b<0)
        printf("Buzz\n");
    else printf("FizzBuzz\n");
}
```

C Code for 1(c)

```
char rank;
scanf("%c", &rank);
int bonus = 0;
switch(rank) {
    case 'p':bonus += 20;
    case 'g':bonus += 20;
    case 's':bonus += 20;
        break;
    default: bonus += 10;
}
printf("\n%d", bonus);
```

C Code for 2(a)

2. (a) **Rewrite** the code segment (see **above right**) using “**if ... else**” without changing the logical meaning. [3]

- (b) **Manually trace** the following code segment and show the change of values of the variables **i, j, n** in each step. [3]

```
#include <stdio.h>
void main() {
    int i = 2, n = 10, j=0;
    for(j = n; j > i; j--) {
        if(j % 2 == 0) i++;
        else n--;
    }
    i += 2;
}
```

3. (a) **Write a C program** that takes an integer **n** as input from the user and **prints** a specific pattern given as follows. For example, for **n = 4**, the output pattern will be as follows. You must program for **n**, NOT for 4. [3]

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

- (b) **Replace** all the “**for**” loops in the following code using only “**while**” loops without changing the logical meaning of the program. [3]

```
int arr[10]= {0};
int k = 15,
for(int i=1; i<6; i+=2){
    arr[i] = ++k-2;
    k++;
}
int c = 0;
```

```
for(int i=6; i<10; i++){
    for(int j=9; j>=i; j--){
        arr[j] = ++c;
    }
for(int i=0; i<10; i++){
    if(i%2==0) arr[i] = ++k;
}
```

4. (a) **Write a C program** that takes **n** number of integers as input into an array of size **N**, where **n** is an odd number and **n<=N**. Your task is to **reverse the first half** array elements and the **last half** array elements, keeping **only the middle element** intact. [3]

Initial Array Elements	Final Array Elements
1 2 3 4 5 6 7	3 2 1 4 7 6 5
10 20 30 40 50	20 10 30 50 40
9 8 7	9 8 7

- (b) **Draw a flow chart** to take an integer as input. Then, display its **odd factors** and calculate the **sum** of its **even factors**. Hint: any integer number is a multiple of any of its factors. [3]

Sample input	Sample output
20	1 5 [Odd factors] 36 [Sum of the even factors: 2 + 4 + 10 + 20 = 36]
28	1 7 [Odd factors] 48 [Sum of the even factors: 2 + 4 + 14 + 28 = 48]

5. (a) **Manually trace** the given code segment. Show the changes of all the variables **i, j, jump**, and array **A** and **B** elements in each step. [3]

```
int A[4]={3, 2, 1};
int B[4]={10, 20, 30};
int jump=100;
for(int i=0; i<3; i++){
    jump = A[i] * 2;
    for(int j=0; j<3; j++){
        B[j] = A[j] + B[j];
        jump = B[j]/2;
    }
    A[i]++;
}
```

C Code for 5(a)

```
int arr[][4]={5, 7, 3, 13},
             {31, 2, 11, 23},
             {17, 19, 43, 53},
             {37, 47, 29, 61}};
int n=4, sum=0, x = 0;
for(int i = 0; i<n; i++){
    for(int j = 0; j<n; j++){
        if(j==n-1 || i+j==n-1){
            x = arr[i][j];
            sum+=x;
        }
    }
}
```

C Code for 5(b)

- (b) **Manually trace** the given code segment (see **above right**) and show the changes of all the variables **i, j, x**, and **sum** in each step. [3]