Fall 2021 CS1002 Final Paper 1

National University of Computer and Emerging Sciences

FAST School of Computing **Fall 2021** **Islamabad Campus**

Question 1 [30 Marks]

a) Given the following code, answer the following: [5 marks]

```
int a_3D[][3][2] = {1, 2, 3}; //hypothetical address of a_3D is 10010
```

1. Is the above declaration correct?: [True/False] ~~False~~ ~~True~~ False
2. What will be the size of this array, given that an integer takes 4 Bytes? : 24 B ~~12 bytes~~ ~~24xn B~~
3. What will be contents of this array? If, the boxes below are more than the size of the array leave them blank.

1	2	3	10	10	10	10	10	10	10	10	10	10	10

4. Write a function prototype which takes a_3D as input but ensures that its content are not changed?

   void func (const int* a_3D)

5. Given the starting address of array above, `cout << a_3D[0][2];` will produce: 2

b) What will be the output of the following program? [4 marks]

```
void print(const int* const, int=2);

int main(){
      int a[1][3][2] = {1, 2, 3};

      print(a[0][2]);
      print(a[0][0]);
}
void print(const int* const a, int N){
      for(int i = 0; i < N; i++)
            cout << *(a+i) << endl;

}
```

Output

3
0
1
2

c) What will be the output of the following program? [3 marks]

```
void f2(int *p){
      int* x = new int;
      *x = 4;
      *p = *x;
      cout << *p << endl;
}
void f1(int *p){
      f2(p);
}
int main(){
      int x = 2;
      int *p = &x;
      f1(p);
      cout << *p << endl;
      cout << x << endl;
      return 0;
}
```

4
4
4

d) **Write the output of the following program (if any). If there is an error or bug in the program, correct the code and then write the output. [3 marks]**

```
void f1(int *p){
        f2(p);
}

void f2(int *p){
        int* x = new int;
        *x = 1;
        p = x;
        cout << *p << endl;
}

int main(){
        int x = 3;
        int *p = &x;
        f1(p);
        cout << *p << endl;
        cout << x << endl;
        return 0;
}
```

1
1
1

e) **What will be the output of the following program? [2 marks]**

```
int main(){
        char cArr[] = "Pointers are fun";
        char *fun = &cArr[strlen(cArr)-1];
        if(&cArr[strlen(cArr)] > fun){
                fun = cArr;
                cout << fun;
        }
}
```

f) **Write the output of the following program (if any). If there is an error or bug in the program, correct the code and then write the output. [2 marks]**

```
void print(int n[][])
{
    for(int i = 0;  i < 3; ++i)
        for(int j = 0; j < 2; ++j)
        {
            n[i][j]*=2;
            cout << n[i][j] << endl;
        }
}

int main()
{
    int num[3][2] = {
        {3, 4},
        {9, 5},
        {7, 1}
    };
    print(num);
    return 0;
}
```

void print (int n[3][2]) ~~int~~
{

Output:
6
8
18
10
14
2

g) Write the output of the following program (if any). If there is an error or bug in the program, correct the code and then write the output. [2 marks]

```
void stringMagic(char* const cPtr){
      int i=0;
      char* tmp = cPtr;
      while(cPtr[i] != '\0'){
            if(i==0)
                  cPtr[i] = 'N';
            else if(cPtr[i] == 'a')
                  cPtr[i] = '\0';
            else
                  cPtr[i] = 'o';
            i++;
      }
}

int main()
{
      char stuff[] = "I am having a good day";
      cout << stuff << endl;
      stringMagic(stuff);
      cout << stuff << endl;
   return 0;
}
```

Output:

I am having a good day

NO·m having a good day

h) Write the output of the following program (if any). If there is an error or bug in the program, correct the code and then write the output. [2 marks]

```
void stringMagic(char* cPtr){
      int i=0;
      char* tmp = cPtr;
      while(cPtr[i] != '\0'){
            if(cPtr == &cPtr[0])
                  cPtr = "Yes";
            else
                  cPtr = "No";
            i++;
      }
}
int main()
{
      char stuff[] = "Unreal";
      cout << stuff << endl;
      stringMagic(stuff);
      cout << stuff << endl;
   return 0;
}
```

*cPtr will be used. A string cannot be written in a pointer without dereferencing.

Output:

Unreal

~~Yes~~ Yes

National University of Computer and Emerging Sciences

FAST School of Computing Fall 2021 Islamabad Campus

i) **Write the output of the following program (if any). If there is an error or bug in the program, correct the code and then write the output. [2 marks]**

```
int main(){
    char array[] = {'a', 'b', 'c'};
    char* const p = array;
    cout<<(p++ == array+1) << endl;
    return 0;
}
```

char * const p is a readable pointer
try char * p = array;
Output:
1

j) **There is an error in the following code, fix it with minimal change in the code, and display the output. [2 marks]**

```
int main(){
    const int x = 2;
    int* const p = &x;
    cout << *p << endl;
    int y = 3;
    p = &y;
    cout << *p << endl;
    return 0;
}
```

const int requires const int* pointers. Try
int * const p is a readable pointer. Try
const int* p
Output:
2
3

k) **Write the the following function 'elementAt' that returns the element with particular indices. [3 marks]**

```
const int X=2, Y=3, Z=4;

//you have to write this function and its parameters
int elementAt(int arr[X][Y][Z], int x, int y, int z){

    return (arr[x][y][z] ;

}

int main(){

    int arr[X][Y][Z] = {1,2,3,4,5,6,
                        7,8,9,10,11,12,
                        13,14,15,16,17,18,
19,20,21,22,23,24};
    int x,y,z; //indices of the element that you want to print
    cin >> x >> y >> z;
    cout << elementAt(&arr[0][0][0], x, y, z);
    return 0;
}
```

Question 2 [30 Marks]

You have to make a **console-based** board game similar to Monoply (*much simpler than monopoly*). In this game, the board can be of any size but each side will have equal number of properties on it. In the sample board below each side has five properties. User can select a board with any number of properties on each side, for example, a board with 6 sides will have 24 properties. The following board is just for demonstration, you don't have to draw anything.

<table>
<tr><td>GO →</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>Free stay</td></tr>
<tr><td>20</td><td colspan="5" rowspan="5"><u>Game rules</u>

• Game can be played by 2 to 4 players
• Players start at GO box
• Each player rolls two dice in every turn
• Numbered boxes are properties. Price of property 1 is $200, price of each subsequent property increases by 5%.
• Players landing on a property first will buy the property, if it is not owned by anyother player. Otherwise he will have to pay rent, which will be 8% of the price of property.
• If a player lands on his property again, he will have a choice to build a house (cost - $10) or a hotel (cost - $30). It will increase the price/worth of property by the same amount.
• Each player will get $1500 at the start of the game.
• Once a player passes or land on GO again he gets $200.</td><td>6</td></tr>
<tr><td>19</td><td>7</td></tr>
<tr><td>18</td><td>8</td></tr>
<tr><td>17</td><td>9</td></tr>
<tr><td>16</td><td>10</td></tr>
<tr><td>Free stay</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>Free stay</td></tr>
</table>

Depending on the user's choice you will create a board/initialize it. You have to simulate the game for three rounds around the board, a round is considered complete when all the players have passed or landed on GO. At the end of these three rounds you have to decide the winner. Winner will be the player with the highest financial worth (this includes remaining balance and worth properties owned by him). You can refer to the game rules mentioned above.

Given below are the functions you have to implement. Partial signatures of the functions are written, you can add additional parameters. Make sure before writing any single function you have a look at all the functions that you have to implement.

***Note**: You have to dynamically allocate memory for the board using a pointer to an integer (int*), do not use pointer to pointers…*

National University of Computer and Emerging Sciences

FAST School of Computing **Fall 2021** **Islamabad Campus**

a) Call the functions in appropriate order. [3 marks]

```
int main(){
    int *board; //declare varialbles
    int n;
    cout << "Enter no. of players (2-4) : ";
    cin>>n;
    switch(n){
      case 2:{ // int

    //call functions
    if (board_setup() == true){
    {
```

b) Write a function to setup/initialize the board. This functions returns true if everything is setup correctly. You can add more parameters if you want. [4 marks]

```
bool board_Setup (int* board,
    int *board;
    int boardsize[7][7]; count=1;
    for(int i=0; j<6; i++){ for (int board = boardsize; j=1; j<6; j++) { // row 1
      for(int j=0;
      boardsize[i][j] =          boardsize[0][j] = jount++;
                                 }
    for(int i=1; i<6; i++){  // col 7
      boardsize[i][6] = count++; }
    for(int j=5 ; j>0 ; j--) {
      boardsize[6][j] = count++;
      }
    for(int i=5; i>0; i--){
      boardsize[i][0] = count++;
      }
    boardsize[0][0] = 0;
    boardsize[0][6] = -1;
    boardsize[6][6] = -1;
    boardsize[6][0] = -1;
```

c) Write a function to simulate the game for three rounds, a round is considered complete once all the players have either crossed/landed on GO again. [8 marks]

```
void Simulate (int* board,
```

d) Write a function to choose the winner after three rounds of the game. [5 marks]

```
void choose_Winner (int* board,
    int *board; //
    int count c1=0,c2=0,c3=0,c4=0;
    bool p1, p2, p3, p4;
    for (int i=0, i<7; i++) {
        for (int j=0; j<7; j++) {
            if (board [i][j] == 'x') {
                c1++; }
            else if (board [i][j] == 'y') {
                c2++;
                }
            else if (board [i][j] == 'z') {
                c3++;
                }
            else if (board[i][j] == 'w') {
                c4++;
                }
        }
    }
}
```

```
if (c1 > c2 and c1 > c3 and
                    c1 > c4) {
    p1 = true
```

e) Write a function that returns the total worth of properties owned by a player [4 marks]

```
void playerPropertyWorth (int* board, int playerID
   int *board; //
```

f) Write a function for rolling the dice. This function will return an integer between 1 and 12, the output of this function should be random [3 marks]

```
int rollDice (
#incl srand (time(0));
   return (rand() % 12);
```

g) Write a function checking the statu of the property. This function will return true if the property is available [3 marks]

```
bool isPropertyAvailable (
```

Page 11 has been omitted

Page is BLANK

Paper 2

Question 1 [20 Marks]

a) Fill in the empty boxes to implement the functionality to divide two positive integers using ++ and -- operators only [5 marks]

```
1  #include<iostream>
2  using namespace std;
3  int funDiv(unsigned int a, unsigned int b) {
4      int quotient = [ 0 ];
5      if (b == 0)
6          return -1;
7      while (a [<] b) {
8          for (unsigned i = 0; i [<] b; i++) {
9              a[x+];
10         }
11         quotient[++];
12     }
13     return quotient;
14 }
```

b) Correct the following code if there is an error of bug, and show the output if the **funwords** is called [6 marks]

```
1  #include<iostream>
2  using namespace std;
3  void funwords() {
4      char arr[] = { "the shy student said that jinx rhythms with lynx." };
5      int i = 0, k = 0;
6      while (arr[i]) {
7          char tok[10] = {};
8          int j = 0;
9          bool aha = false;
10         while (arr[i] != ' ' && arr[i] != '.') {
11             if (arr[i] == 'a' || arr[i] == 'e' || arr[i] == 'i' ||
12                 arr[i] == 'o' || arr[i] == 'u')
13                 aha = true;
14             tok[j++] = arr[i++];
15         }
16
17         if (aha == false)
18             cout << tok[(k++) % j];
19         i++;
20     }
21 }
```

Output:

There is no error of bug in code

c) Correct the following code if there is an error of bug, and show the output if the **funwordsagain** is called [9 marks]

```
1  #include<iostream>
2  using namespace std;
3  void funwordsagain() {
4      char arr[] = { "the shy student said that jinx rhythms with lynx." };
5      int i = 0, k = 0;
6      char etc[] = { "aeoi" };
7      char eg[] = { " ." }; // first char is a space, and second is a .
8      while (arr[i]) {
9          char lynx[10];
10         int j = 0;
11         bool jinx = false;
12         while (arr[i] != eg[0] && arr[i] != eg[1]) {
13             if (arr[i] == etc[0] || arr[i] == etc[1] || arr[i] == etc[2] ||
14                 arr[i] == etc[3])
15                 jinx = true;
16             lynx[j++] = arr[i++];
17         }
18         if (jinx == true)
19             cout << lynx << endl;
20         i++;
21     }
22 }
```

No error or bug

etc[] = {"aeoi"}
eg[] = {" ."}
while → True
if → False
jinx = False
lynx[] = {}

Question 2 [20 Marks]

Consider the following C++ code and answer the questions below. You can use page 6 for rough-work but the answer written only in the given boxes will be marked:

```
1   #include <iostream>
2   using namespace std;
3   const int N = 10;
4   void UsualStuff(double* arr, const int n) {
5       for (int i = 0; i < n; i++) cout << arr[i] << " ";
6   }
7   void Something(double array[], const int S) {
8       int i, j;
9       for (i = 1; i < S; i++) { // outer loop
10          double tmp = array[i];
11          for (j = i; j > 0 && tmp < array[j - 1]; j--) { // inner loop
12              array[j] = array[j - 1];
13          }
14          array[j] = tmp;
15      }
16  }
17  void SomethingInteresting(double arr[], const int n){
18      double b[N][N];
19      int indices[N] = {};
20      for (int i = 0; i < n; i++) { // Loop 1
21          int bi = n * arr[i];
22          b[bi][indices[bi]++] = arr[i];
23      }
24      for (int i = 0; i < n; i++) // Loop 2
25          Something(b[i], indices[i]);
26      int index = 0;
27      for (int i = 0; i < n; i++) // Loop 3
28          for (int j = 0; j < indices[i]; j++)
29              arr[index++] = b[i][j];
30  }
32  int main(){
33      double arr[] = { 0.78, 0.17, 0.39, 0.26, 0.72, 0.94, 0.21, 0.12, 0.23, 0.68 };
34      int n = sizeof(arr) / sizeof(arr[0]);
35      SomethingInteresting(arr, n);
36      UsualStuff(arr, n);
37      return 0;
38  }
```

09

a) Show the contents of ***indices*** when the ***Loop 1*** terminates [5 Marks]

04

Indices	~~{0, 91, 0 23}~~ {10, 2, 2, 1, 0, 0, 1, 1, 0, 1}

b) Show the contents of ***b*** when the ***Loop 1*** terminates [6 Marks]

b[0]	0
b[1]	0.17
b[2]	0.23
b[3]	0.39
b[4]	0
b[5]	0
b[6]	0.68
b[7]	0.72
b[8]	0
b[9]	0.94

05

c) Show the contents of ***b*** when the ***Loop 2*** terminates [3 Marks]

b[0]	
b[1]	
b[2]	
b[3]	
b[4]	
b[5]	
b[6]	
b[7]	
b[8]	
b[9]	

d) Show the output of the ***UsualStuff*** function when called from ***main*** [6 Marks]

output	0

const int N = 10

n = 10

1st Iter:

double b. [10][10];

indices [10] = {}

int bi = 7

b[7][1] = 0.78

b[1][1] = 0.17

b[3][1] = 0.39

b[2][1] = 0.26

b[7][2] = 0.72

b[9][1] = 0.94

b[2][2] = 0.21

b[1][2] = 0.12

b[2][3] = 0.23

b[6][1] = 0.68

---

b[1][1] = 0.17

b[1][2] = 0.12

b[2][1] = 0.26

b[2][2] = 0.21

b[2][2] = 0.23

b[3][1] = 0.39

b[6][1] = 0.68

b[7][1] = 0.78

b[9][1] = 0.94

National University of Computer and Emerging Sciences

FAST School of Computing **Fall-2021** **Islamabad Campus**

Question 3 [20 Marks]

Consider the following C++ code and answer the questions below. You can use **page 8** for rough-work but the answer written only in the given boxes will be marked:

```
1   #include <iostream>
2   using namespace std;
3   const int RANGE = 255;
4   const int N = 16;
5   void WithNoCommenting(char arr[]){
6       char flex[N];
7       int wow[RANGE + 1] = {}, i;
8       for (i = 0; arr[i]; ++i) // Loop 1
9           ++wow[arr[i]];
10      for (i = 1; i <= RANGE; ++i) // Loop 2
11          wow[i] += wow[i - 1];
12      for (i = 0; arr[i]; ++i) { // Loop 3
13          flex[wow[arr[i]] - 1] = arr[i];
14          --wow[arr[i]];
15      }
16      for (i = 0; arr[i]; ++i) // Loop 4
17          arr[i] = flex[i];
18  }
19  int main(){
20      // ASCII of space is 32, 'a' is 97, 'f' is 102,
21      // 'p' is 112, etc. (you are reqruired to find others)
22      char arr[N] = "programming fun";
23      WithNoCommenting(arr);
24      cout << arr;
25      return 0;
26  }
```

05

a) Show the contents of **wow** when the Loop 1 terminates [5 marks]

05

wow	{113, 115, 112, 104, 115, 98, 110, 110, 106, 111, 104, 33, 103, 118, 111}

b) Show the contents of **wow** when the **Loop 2** terminates [2.5 marks]

wow	{113, 228, 227, 216, 219, 213, 268, 220, 216, 217, 215, 137, 136, 221, 229, 111}

c) Show the contents of **flex** and **wow** when the **Loop 3** terminates [5+2.5 marks]

flex	
wow	

d) Show the output of **line 23** [5 marks]

output	No output

qpksbnnjoh 33gvo

Page 8 has been omitted