CS101 REVISION SESSION

Autumn 2024-25
(Preetish, Neel, Latha)

1-D Arrays

Q1 Find the error in this code (Easy)

```
int main() {
    int arr[5];
    for(int i = 1; i <= 5; i++) {
        arr[i] = i * 2;
    }
    for(int i = 1; i <= 5; i++) {
        cout << arr[i] << " ";
    }
    return 0;
}
```

```
int main() {
    int arr[] = {1, 2, 3, 4, 5};
    int n = sizeof(arr) / sizeof(arr[0]);
    int sum = 0;
    for(int i = 0; i < n; i++) {
        sum += arr[n];
    }
    cout << "Sum: " << sum << endl;
    return 0;
}
```

Q1 Find the error in this code (Easy)

```
int main() {
    int arr[5];
    for(int i = 0; i < 5; i++) {
        arr[i] = i * 2;
    }
    for(int i = 0; i < 5; i++) {
        cout << arr[i] << " ";
    }
    return 0;
}
```

```
int main() {
    int arr[] = {1, 2, 3, 4, 5};
    int n = sizeof(arr) / sizeof(arr[0]);
    int sum = 0;
    for(int i = 0; i < n; i++) {
        sum += arr[i];
    }
    cout << "Sum: " << sum << endl;
    return 0;
}
```

Q2 Find the error in this code (Medium)

```
int main() {
    int arr[] = {1, 2, 3, 4, 5};
    int n = sizeof(arr) / sizeof(arr[0]);
    int k = 2;
    for(int i = 0; i < n; i++) {
        arr[i] = arr[(i + k) % n];
    }
    for(int i = 0; i < n; i++) {
        cout << arr[i] << " ";
    }
    return 0;
}
```

Here you need to rotate a given array left by k.

Q2 Find the error in this code (Medium)

```
int main() {
    int arr[] = {1, 2, 3, 4, 5};
    int n = sizeof(arr) / sizeof(arr[0]);
    int k = 2;
    int temp[n];
    for(int i = 0; i < n; i++) {
        temp[i] = arr[(i + k) % n];
    }
    for(int i = 0; i < n; i++) {arr[i] = temp[i];}
    for(int i = 0; i < n; i++) {
        cout << arr[i] << " ";
    }
    return 0;
}
```

Here you need to rotate a given array left by k.

Q3 Output of the given program

```
int main() {
    int arr[] = {10, 20, 30, 40, 50};
    int n = sizeof(arr) / sizeof(arr[0]);
    for(int i = 0; i < n; i++) {
        arr[i] = arr[i] + arr[(i + 1) % n];
    }
    for(int i = 0; i < n; i++) {
        cout << arr[i] << " ";
    }
    return 0;
}
```

Q3 Output of the given program

```
int main() {
    int arr[] = {10, 20, 30, 40, 50};
    int n = sizeof(arr) / sizeof(arr[0]);
    for(int i = 0; i < n; i++) {
        arr[i] = arr[i] + arr[(i + 1) % n];
    }
    for(int i = 0; i < n; i++) {
        cout << arr[i] << " ";
    }
    return 0;
}
```

Output :

```
30 50 70 90 80
```

Q4 Fill in the blanks - Find the kth missing number

```
int findkmissing(vector<int> &nums, int k){
    int low = 0, high = nums.size()-1;
    int mid = (low+high)/2;
    while(low <= high){
        mid = (low+high)/2;
        if(nums[mid]-(mid+1)<k and nums[mid+1]-(mid+2)>=k){cout<<k+mid+1;return;}
        else if(nums[mid]-(mid+1)>=k){high = mid-1;}
        else{low = mid+1;}
    }
    cout<<k+mid+1;
}
```

Q4 Fill in the blanks - Find the kth missing number

```
int findkmissing(vector<int> &nums, int k){
    int low = 0, high = nums.size()-1;
    int mid = (low+high)/2;
    while(low <= high){
        mid = (low+high)/2;
        if(______ and ______){cout<<______;return;}
        else if(______){high = mid-1;}
        else{low = mid+1;}
    }
    cout<<_____;
}
```

Vectors, Maps and Pairs

Q1 Find output of the given code

```
void func(const vector<int>& arr, int k) {
    map<int, int> abcd;
    for (int num : arr) {
        abcd[num]++;
    }
    for (auto pair : abcd) {
       if (pair.second >= k) {cout << pair.first << " "; }
    }
}
```

```
int main() {
    vector<int> arr = {1, 2, 2, 3, 3,
3, 4, 4, 4, 4};
    int k = 2;
    func(arr, k);
    return 0;
}
```

Q1 Find output of the given code

```
void func(const vector<int>& arr, int k) {
    map<int, int> abcd;
    for (int num : arr) {
        abcd[num]++;
    }
    for (auto pair : abcd) {
       if (pair.second >= k) {cout << pair.first << " "; }
    }
}
```

Output :

```
2 3 4
```

Q2 Fill in the blanks (Easy)

```
int main() {
    vector<pair<int, int>> v = {____(5, 6)};
    v.____({1, 5});
    vector<int> sums;
    for (____ p : v) {
        sums.push_back(p.____ + p.____);
    }
```

```
cout << "Sums of first and second elements: ";
    for (___ sum : sums) {
        cout << sum << " ";
    }
    cout << endl;
    return 0;
}
```

Find the sum of each pair and store it in another vector

Q2 Fill in the blanks (Easy)

```
int main() {
    vector<pair<int, int>> v = {make_pair(5, 6)};
    v.push_back({1, 5});
    vector<int> sums;
    for (auto p : v) {
        sums.push_back(p.first + p.second);
    }
```

```
cout << "Sums of first and second elements: ";
    for (auto sum : sums) {
        cout << sum << " ";
    }
    cout << endl;
    return 0;
}
```

Find the sum of each pair and store it in another vector

Q3 Find the error in the given code

Given a vector of pairs, with the first element being a number and the second element is a colour. Sort the array by colour, preserving the order of the first element.

```
void solve(){
    int n, i = 0;
    vector<int, string> v;


    for(int i = 0; i<n; i++){
        int a, s;
        cin>>a>>s;
        v.push_back(a, s);
    }
    map<string, vector<int>> m;
    for(auto p : v){
        m[p.first].push_back(p.second);
    }


    for(auto a : m){
        for(auto b : a.first){
            cout<< b << " " << a.second << endl;
        }
    }
}
```

Q3 Find the error in the given code

Given a vector of pairs, with the first element being a number and the second element is a colour. Sort the array by colour, preserving the order of the first element.

```
void solve(){
    int n, i = 0;
    vector<pair<int, string>> v;
    cin>>n;
    for(int i = 0; i<n; i++){
        int a;    string s;
        cin>>a>>s;
        v.push_back({a, s});
    }
    map<string, vector<int>> m;
    for(auto p : v){
        m[p.second].push_back(p.first);
    }

    for(auto a : m){
        for(auto b : a.second){
            cout<< b << " " << a.first << endl;
        }
    }
}
```

```
 void allocateMem(int **p, int size){
     *p = new int[size];
     for (int i = 0; i < size; i++)
         (*p)[i] = i * 10;
 }

 void deallocateMem(int *p) {
     delete[] p;
     p = nullptr;
 }

int main() {
    int *arr = nullptr;
    allocateMem(&arr, 5);
    cout << arr[3] << endl;  ----(1)
    deallocateMem(arr);
    cout<<(arr == nullptr?"Null":"Not Null")<<endl;  ----(2)
    return 0;
}
```

What will be the output at (1)?

What will be the output at (2)?

```
 void allocateMem(int **p, int size){
     *p = new int[size];
     for (int i = 0; i < size; i++)
         (*p)[i] = i * 10;
 }

 void deallocateMem(int *p) {
     delete[] p;
     p = nullptr;
 }
int main() {
    int *arr = nullptr;
    allocateMem(&arr, 5);
    cout << arr[3] << endl;   ----(1)
    deallocateMem(arr);
    cout<<(arr == nullptr?"Null":"Not Null")<<endl;  ----(2)
    return 0;
}
```

What will be the output at (1)?

30

What will be the output at (2)?

Not Null (why??)

Follow Up: Modify code to make arr a Null Pointer?

```
 void allocateMem(int **p, int size){
     *p = new int[size];
     for (int i = 0; i < size; i++)
         (*p)[i] = i * 10;
 }

 void deallocateMem(int *&p) {
     delete[] p;
     p = nullptr;
 }
int main() {
    int *arr = nullptr;
    allocateMem(&arr, 5);
    cout << arr[3] << endl; ----(1)
    deallocateMem(arr);
    cout<<(arr == nullptr?"Null":"Not Null")<<endl; ----(2)
    return 0;
}
```

Follow Up: Modify code to make arr a Null Pointer?

Pass the pointer reference of arr

Inserting element in a sorted linked list:

```
struct Node{
  int data;
  Node* next;
  Node(int v): data(v), next(nullptr){}
};
```

```
void insertNode(Node** head, int val){
     Node* newNode = new Node(val);

     //Case 1: If head is null or head->data == value
     if(____(a)____){
          newNode->next = *head;
          head = newNode;
          return;}

     //Case 2: Find correct position
     Node* curr = *head;
     while(____(b)____) curr = curr->next;

     ____(c)____;
     ____(d)____;
}
```

Fill in for a,b,c,d

Inserting element in a sorted linked list:

```
struct Node{
  int data;
  Node* next;
  Node(int v): data(v), next(nullptr){}
};
```

```
void insertNode(Node** head, int val){
     Node* newNode = new Node(val);

     //Case 1: If head is null or head->data == val
     if(*head==nullptr || (*head)->data==val){
          newNode->next = *head;
          head = newNode;
          return;}

     //Case 2: Find correct position
     Node* curr = *head;
     while(curr->next && curr->next->data < val) curr = curr->next;

     ____(c)____;
     ____(d)____;
}
```

Inserting element in a sorted linked list:

```
struct Node{
  int data;
  Node* next;
  Node(int v): data(v), next(nullptr){}
};
```

```
void insertNode(Node** head, int val){
     Node* newNode = new Node(val);

     //Case 1: If head is null or head->data == val
     if(*head==nullptr || (*head)->data==val){
          newNode->next = *head;
          head = newNode;
          return;}

     //Case 2: Find correct position
     Node* curr = *head;
     while(curr->next && curr->next->data < val) curr = curr->next;

     newNode->next = curr->next;
     curr->next = newNode;
}
```

Dice Combinations:

Count the number of ways to construct sum n (>0) by throwing a dice one or more times. Each throw produces an outcome between 1 and 6.

For example, if n=3, there are 4 ways:

- 1+1+1
- 1+2
- 2+1
- 3

Hint: Try recursion

```
int func(int n){

     //Write your function here

}
```

Dice Combinations:

Count the number of ways to construct sum n (>0) by throwing a dice one or more times. Each throw produces an outcome between 1 and 6.

For example, if n=3, there are 4 ways:

- 1+1+1
- 1+2
- 2+1
- 3

```
int func(int n){
    if(n < 0)
    return 0;

    if(n == 0)
    return 1;

    return func(n-1)+func(n-2)+
           func(n-3)+func(n-4)+
           func(n-5)+func(n-6);
}
```

HW: This will take too much time. Can you think of a better approach?

Binary to Gray Code conversion:

Given a binary string, convert it into its equivalent Gray Code.

Binary: 0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111
Gray: 0000 0001 0011 0010 0110 0111 0101 0100 1100 1101 1111 1110 1010 1011 1001 1000

Eg: Input: 11 □ 10

Hint: Use Recursion
Analyse last 2 digits of input.

```
int func(int n){

     //Write your func here

}
```

Binary to Gray Code conversion:

Given a binary string, convert it into its equivalent Gray Code.

Binary: 0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111
Gray: 0000 0001 0011 0010 0110 0111 0101 0100 1100 1101 1111 1110 1010 1011 1001 1000

Fill in code snippets a,b

```
int func(int n){
     if(n == 0)
     return;

     int a = n%10;
     int b = (n/10)%10;

   //If last 2 digits are opposite
   ____(a)____
   //If last 2 digits are same
   ____(b)____
}
```

Binary to Gray Code conversion:

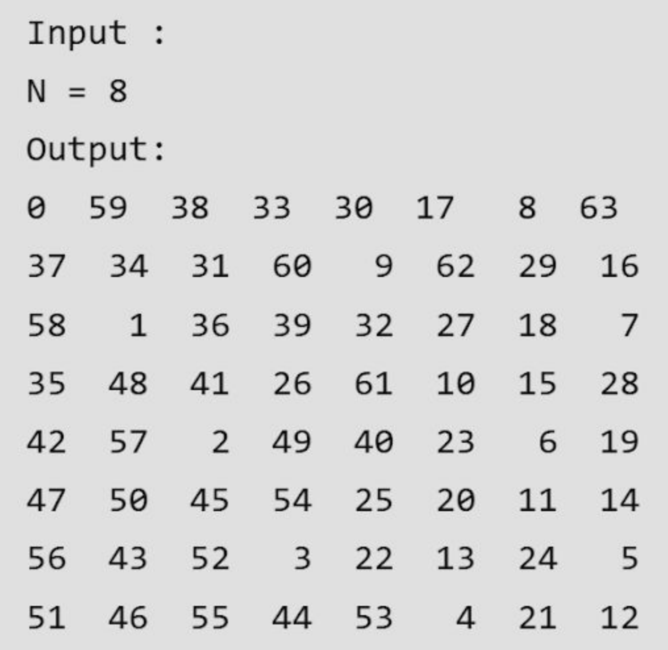
Given a binary string, convert it into its equivalent Gray Code.

Binary: 0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111
Gray: 0000 0001 0011 0010 0110 0111 0101 0100 1100 1101 1111 1110 1010 1011 1001 1000

```
int func(int n){
   if(n == 0)
   return 0;

   int a = n%10;
   int b = (n/10)%10;

   //If last 2 digits are opposite
   if(a==1 && b==0 || a==0 && b==1)
   return 1 + func(n/10)*10;

   //If last 2 digits are same
   return func(n/10)*10;
}
```

Knight’s Tour:

Given a N*N board with the Knight placed on the first block of an empty board. Moving according to the rules of chess knight must visit each square exactly once. Print the order of each cell in which they are visited.



```
Input :
N = 8
Output:
0  59  38  33  30  17   8  63
37  34  31  60   9  62  29  16
58   1  36  39  32  27  18   7
35  48  41  26  61  10  15  28
42  57   2  49  40  23   6  19
47  50  45  54  25  20  11  14
56  43  52   3  22  13  24   5
51  46  55  44  53   4  21  12
```

```
vector<vector<int>> sol(n, vector<int> (n,
-1));
//initialize each element of sol to -1

int helper(int x, int y, int movn);

void solv(int n){
  sol[0][0] = 0; //Start Position

  if(helper(0,0,1) == 0)
  cout<<“No solution”<<endl;
  else
  //print sol
}
```

Knight’s Tour:

```
int xMove[8] = {2,1,-1,-2,-2,-1,1,2};
int yMove[8] = {1,2,2,1,-1,-2,-2,-1};

int sol[n][n];
//initialize each element of sol to -1

int helper(int x, int y, int movn);

void solv(int n){
  sol[0][0] = 0; //Start Position

  if(helper(0,0,1) == 0)
  cout<<“No solution”<<endl;
  else
  //print sol
}

int isValid(x,y){
  return (x>=0 && y>=0 && x<N && y<N &&
          sol[x][y] == -1);
}
```

```
int helper(int x, int y, int movn){
   if(movn == N*N)
   return 1;

   for(int k = 0; k < 8; k++){
      int x_n = x+xMove[k];
      int y_n = y+yMove[k];

      if(isValid(x_n,y_n)){
         sol[x_n][y_n] = movn+1;
         if(helper(x_n,y_n,movn+1)
            return 1;
         else
            sol[x_n][y_n] = -1;
      }
   }
   return 0;
}
```

Question 1

Find the errors

```
class Car{
        string brand;
        int year;
        double price;
public:
        Car(string b, int y, double p) :
                    brand(b), year(y), price(p);
        void display() {
                cout << brand << " " << year
                        << " " << price << endl;
        }
};
int main () {
        Car car1;
        car1("Toyota", 2019, 30000);
        car1.display();
}
```

```
public:
        Car(string b, int y, double p) : brand(b), year(y), price(p);
```

Constructor should be defined somewhere inside or outside the class

The above is just a declaration!

Fix:

```
public:
        Car(string b, int y, double p) : brand(b), year(y), price(p) { }
```

```
int main () {
        Car car1;
        car1("Toyota", 2019, 30000);
        car1.display();
}
```

```
Car car1;
```

==> Car class has no constructor with zero arguments

```
car1("Toyota", 2019, 30000);
```

==>Cannot call object as a function.

Correct Code:

```
class Car{
    string brand;
    int year;
    double price;
public:
    Car(string b, int y, double p) : brand(b), year(y), price(p){}
    void display() {
        cout << brand << " " << year << " " << price << endl;
    }
};
int main () {
    Car car1("Toyota", 2019, 30000);
    car1.display();
}
```

Question 2

Find Output

```
class Demo {
        int* data;
public:
        Demo(int value) {
                data = new int;
                *data = value;
                cout << "Created Object with data:  " << *data << endl;
        }
        ~Demo() {
                cout << *data << " : "   ;
                delete data;
                cout << " Memory Deallocated "  <<endl;
        }
        void display () {
                cout << "Displaying data: " <<*data << endl;
        }
};
```

```
int main() {
        Demo obj1(20);
        obj1.display();

        Demo* obj2 = new Demo(20);
        obj2->display();

        delete obj1;
        delete obj2;

        cout << "Exiting…… "  << endl;
        return 0;
}
```

The above code gives error.why???

The above code gives error.why???

delete obj1;

==> delete works only for manually created object

```
class Demo {
    int* data;
public:
    Demo(int value) {
        data = new int;
        *data = value;
        cout << "Created Object with data:  " <<
    *data << endl;
    }
    ~Demo() {
        cout << *data << “ : ” ;
        delete data;
        cout << " Memory Deallocated "  <<endl;
    }
    void display () {
        cout << "Displaying data:  " <<*data <<
    endl;
    }
};
```

```
int main() {
    Demo obj1(20);
    obj1.display();

    Demo* obj2 = new Demo(10);
    obj2->display();

    delete obj2;

    cout << "Exiting……. "  << endl;
    return 0;
}
```

Output:

```
Created Object with data 20
Displaying data: 20
Created Object with data 10
Displaying data: 10
10 :  Memory Deallocated
Exiting…….
20 :  Memory Deallocated
```

Question 3

Find Output

```
class Complex {
        int real;
        int img;
public:
        Complex(int r=0, int i=0) : real(r), img(i) {}
        Complex operator+(const Complex& other)  const {
                return Complex(real+other.real, img+other.img);
        }
        bool operator==(const Complex& other) const {
                return ((real==other.real) && (img==other.img));
        }
        void display() {
                cout << real << " " <<  img << endl;
        }
};
```

```
int main() {
        Complex c1(3,4);
        Complex c2(1,2);

        Complex c3 = c1+c2;

        c3.display();
        bool equal = c1==c2;

        if (c1==c2) {
                cout << "Equal" << endl;
        }
        else {
                cout<<"Not equal"<<endl;
        }
}
```

Output:

```
4 6

Not equal
```

Question 4

Fill in the blanks

```
template < typename T1, typename T2>

class Pair {

        T1 first; T2 second;

public:

        Pair (___ a, ___ b) : first(a), second(b) {}

        ___ getfirst() const {

                return first;

        }

        ___ getsecond() const {

                return second;

        }

};
```

```
typedef Pair <___,___> SIpair;

int main() {
        Pair <int,int> obj(1,1);

        cout << obj.getfirst()<< endl;
        cout << obj.getsecond() << endl;

        SIpair p( "abc", 1);

        cout << p.getfirst()<< endl;
        cout << p.getsecond() << endl;

}
```

```
template < typename T1, typename T2>

class Pair {

        T1 first; T2 second;

public:

        Pair (T1 a, T2 b) : first(a), second(b) {}

        ____ getfirst() const {

                return first;

        }

        ____ getsecond() const {

                return second;

        }

};
```

```
typedef Pair <____,____> SIpair;

int main() {
        Pair <int,int> obj(1,1);

        cout << obj.getfirst()<< endl;
        cout << obj.getsecond() << endl;

        SIpair p( "abc", 1);

        cout << p.getfirst()<< endl;
        cout << p.getsecond() << endl;

}
```

```
template < typename T1, typename T2>

class Pair {

    T1 first; T2 second;

public:

    Pair (T1 a, T2 b) : first(a), second(b) {}

    T1 getfirst() const {

        return first;

    }

    T2 getsecond() const {

        return second;

    }

};
```

```
typedef Pair <___,___> SIpair;

int main() {
    Pair <int,int> obj(1,1);

    cout << obj.getfirst()<< endl;
    cout << obj.getsecond() << endl;

    SIpair p( "abc", 1);

    cout << p.getfirst()<< endl;
    cout << p.getsecond() << endl;

}
```

```
template < typename T1, typename T2>
class Pair {
    T1 first; T2 second;
public:
    Pair (T1 a, T2 b) : first(a), second(b) {}
    T1 getfirst() const {
        return first;
    }
    T2 getsecond() const {
        return second;
    }
};
```

```
typedef Pair <string,int> SIpair;

int main() {
    Pair <int,int> obj(1,1);

    cout << obj.getfirst()<< endl;
    cout << obj.getsecond() << endl;

    SIpair p( "abc", 1);

    cout << p.getfirst()<< endl;
    cout << p.getsecond() << endl;

}
```

Question 5

Fill in the blanks

```
class Stack {
    map<int,int> stack;
    int topindex;
public:
    Stack() : __________ {}
    void push(int value);
    int pop();
    int peek();
};
void Stack:: push(int value) {
        __________________;
}
```

```
int Stack:: pop() {
    if (topindex == -1) {
        return -1;
    }
    else {
        int value = stack[topindex];
        stack.erase(____________);
        return value;
    }
}
int Stack:: peek () {
    if (topindex == -1) {
        return -1;
    }
    else {
        return  _______________;
    }
}
```

```
class Stack {
    map<int,int> stack;
    int topindex;
public:
    Stack() : topindex(-1) {}
    void push(int value);
    int pop();
    int peek();
};
void Stack:: push(int value) {
        ____________________;
}
```

```
int Stack:: pop() {
    if (topindex == -1) {
        return -1;
    }
    else {
        int value = stack[topindex];
        stack.erase(_____________);
        return value;
    }
}
int Stack:: peek () {
    if (topindex == -1) {
        return -1;
    }
    else {
        return ________________;
    }
}
```

```
class Stack {
    map<int,int> stack;
    int topindex;
public:
    Stack() : topindex(-1) {}
    void push(int value);
    int pop();
    int peek();
};
void Stack:: push(int value) {
        stack[++topindex] = value;
}
```

```
int Stack:: pop() {
    if (topindex == -1) {
        return -1;
    }
    else {
        int value = stack[topindex];
        stack.erase(______________);
        return value;
    }
}
int Stack:: peek () {
    if (topindex == -1) {
        return -1;
    }
    else {
        return  ________________;
    }
}
```

```
class Stack {
    map<int,int> stack;
    int topindex;
public:
    Stack() : topindex(-1) {}
    void push(int value);
    int pop();
    int peek();
};
void Stack:: push(int value) {
        stack[++topindex] = value;
}
```

```
int Stack:: pop() {
    if (topindex == -1) {
        return -1;
    }
    else {
        int value = stack[topindex];
        stack.erase(topindex--);
        return value;
    }
}
int Stack:: peek () {
    if (topindex == -1) {
        return -1;
    }
    else {
        return ___________;
    }
}
```

```
class Stack {
    map<int,int> stack;
    int topindex;
public:
    Stack() : topindex(-1) {}
    void push(int value);
    int pop();
    int peek();
};
void Stack:: push(int value) {
        stack[++topindex] = value;
}
```

```
int Stack:: pop() {
    if (topindex == -1) {
        return -1;
    }
    else {
        int value = stack[topindex];
        stack.erase(topindex--);
        return value;
    }
}
int Stack:: peek () {
    if (topindex == -1) {
        return -1;
    }
    else {
        return stack[topindex];
    }
}
```