Unit - III **Arrays and Structures** CHAPTER - 12

ARRAYS

PART – I

Choose the correct answer.

1. Which of the following is the collection of variables of the same type that an referenced by a common name?
   a) int b) float **c) Array** d) class
2. Array subscripts is always starts with which number?
   a)-1 **b) 0** c) 2 d) 3
3. int age[]={6,90,20,18,2}; How many elements are there in this array?
   a) 2 **b) 5** c) 6 d) 4
4. cin>>n[3]; To which element does this statement accepts the value?
   a) 2 **b) 3** c) 4 d) 5
5. By default, the string and with which character?
   a)\o b) \t c) \n d) \b

Part – II

Answers to all the questions (2 Marks):

1. What is Traversal in an Array?

Accessing each element of an array at least once to perform any operation is known as “Traversal”. Displaying all the elements in an array is an example of “traversal”.

2. What is Strings?

- A string is defined as a sequence of characters where each character may be a letter, number or a symbol.
- Each element occupies one byte of memory.
- Every string is terminated by a null (‘\0’, ASCII code 0) character

3. What is the syntax to declare two – dimensional array.

- Two-dimensional (2D) arrays are collection of similar elements where the elements are stored in certain number of rows and columns.
- An example **m × n** matrix where ***m*** *denotes the number of rows* and ***n*** *denotes the number of columns.*

Example:

```
int arr[3][3];
```

Part – III

Answers to all the questions (3 Marks):

1. Define an Array? What are the types?

"An array is a collection of variables of the same type that are referenced by a common name". There are different types of arrays used in C++. They are:

1. One-dimensional arrays
2. Two-dimensional arrays
3. Multi-dimensional arrays

2. With note an Array of strings.

An array of strings is a two-dimensional character array. The size of the first index (rows) denotes the number of strings and the size of the second index (columns) denotes the maximum length of each string.

Declaration of 2D Array:

```
char Name[6][10];
```

Initialization:

```
char Name[6][10] = {"Vijay", "Raji", "Suji", "Joshini", "Murugan", "Mani"};
```

3. Write a C++ program to accept and print your name?

```
# include <iostream>
# include <string>
using namespace std;
int main()
{
char myname[30];
cout << " Enter your Name: ";
cin >> myname;
cout<< "\n My name is :" <<myname;
getch();
}
```

Output:

Enter your Name: Joshini

My name is : Joshini

Part – IV

Answer to all the questions (5 Marks):

1. Write a C++ program to find the difference between two matrix.

```
#include<iostream>
using namespace std;
int main()
{
        clrscr();
        int arr1[3][3], arr2[3][3], arr3[3][3], sub, i, j;
        cout<<"Enter 3*3 Array 1 Elements : ";
```

```
        for(i=0; i<3; i++)
        {
                for(j=0; j<3; j++)
                {
                        cin>>arr1[i][j];
                }
        }
        cout<<"Enter 3*3 Array 2 Elements : ";
        for(i=0; i<3; i++)
        {
                for(j=0; j<3; j++)
                {
                        cin>>arr2[i][j];
                }
        }
        cout<<"Subtracting array (array1-array2) ... \n";
        for(i=0; i<3; i++)
        {
                for(j=0; j<3; j++)
                {
                        arr3[i][j]=arr1[i][j]-arr2[i][j];
                }
        }
        cout<<"Result of Array1 - Array2 is :\n";
        for(i=0; i<3; i++)
        {
                for(j=0; j<3; j++)
                {
                        cout<<arr3[i][j]<<" ";
                }
                cout<<"\n";
        }
        getch();
}
```


Output:

```
Enter 3*3 Array 1 Elements : 4
5
6
4
5
6
4
5
6
Enter 3*3 Array 2 Elements : 1
2
3
1
2
3
1
2
3
Subtracting array (array1-array2) ...
Result of Array1 - Array2 is :
3 3 3
3 3 3
3 3 3
```

2. How will you pass two dimensional array to a function explain with example.

- In C++, arrays can be passed to a function as an argument. To pass an array to a function in C++, the function needs the array name as an argument.
- Passing a two-dimensional array to a function

Example: ***C++ program to display values from two dimensional array***

```
#include <iostream>
using namespace std;
void display (int n[3][2]);
int main()
{
int num[3][2] = { {3, 4}, {9, 5}, {7, 1} };
display(num);
return 0;
}
void display(int n[3][2])
{
cout << "\n Displaying Values" << endl;
for (int i=0; i<3; i++)
{
for (int j=0; j<2; j++)
{
cout << n[i][j] << " ";
}
cout << endl << endl;
}
getch( );
}
```

Output:

```
Displaying Values
3  4
9  5
7  1
```

In the above program, the two-dimensional array num is passed to the function display() to produce the results.

STRUCTURES

Part – I

Choose the correct answer

1. The data elements in the structure are also known as

(a) objects (b) members (c) data (d) records

2. Structure definition is terminated by

(a) : (b) } **(c) ;** (d) ::

3. What will happen when the structure is declared?

(a) it will not allocate any memory **(b) it will allocate the memory**

(c) it will be declared and initialized (d) it will be only declared

4. What is the output of this program?

```
#include <iostream>
#include <string.h>
using namespace std;
int main()
{
struct student
{
int n;
char name[10];
};
student s;
s.n = 123;
strcpy(s.name, "Balu");
cout<<s.n;
cout<< s.name <<endl;
return 0; }
```

(a) 123Balu (b)BaluBalu (c) Balu123 (d) 123 Balu

5. A structure declaration is given below.

```
struct Time
{
int hours;
int minutes;
int seconds;
}t;
```

Using above declaration which of the following refers to seconds.

(a) Time.seconds (b) Time::seconds (c)seconds **(d) t. seconds**

6. What will be the output of this program?

```
#include <iostream>
using namespace std;
struct ShoeType
{
string name;
double price;
};
int main()
{
ShoeType shoe1, shoe2;
shoe1.name = "Adidas";
shoe1.price = 9.99;
```

```
cout<< shoe1.name<< " # "<< shoe1.price<<endl;
shoe2 = shoe1;
shoe2.price = shoe2.price / 9;
cout<< shoe2.name<< " # "<< shoe2.price;
return 0;
```

(a) Adidas # 9.99 **Adidas # 1.11**	(b) Adidas # 9.99 Adidas # 9.11	(c) Adidas # 9.99 Adidas # 11.11	(d) Adidas # 9.11 Adidas # 11.11

7. Which of the following is a properly defined structure?

(a) struct {int num;} (b) struct sum {int num;}

(c) struct sum int sum; **(d) struct sum {int num;};**

8. A structure declaration is given below.

```
struct employee
{
int empno;
char ename[10];
}e[5];
```

Using above declaration which of the following statement is correct.

(a) cout<<e[0].empno<<e[0].ename; (b) cout<<e[0].empno<<ename;

(c) cout<<e[0]->empno<<e[0]->ename; (d) cout<<e.empno<<e.ename;

9. Which of the following cannot be a structure member?

(a) Another structure (b) Function (c) Array (d) variable of double datatype

10. When accessing a structure member ,the identifier to the left of the dot operator is the name of

(a) structure variable (b) structure tag (c) structure member (d) structure function

Part – II

Answer to all the questions (2 Marks):

1. Define structure .What is its use?

- Structure is a user-defined which has the combination of data items with different data types.

Use:

This allows to group of variables of mixed data types together into a single unit.

2. To store 100 integer number which of the following is good to use? Array or Structure State the reason.

- Structure is better than array.
- Because Structure variable allocates memory for that variable itself.

3. What is the error in the following structure definition.

```
struct employee{ inteno;charename[20];char dept;}

Employee e1,e2;
```

```
struct employee
{
int eno;
char ename[20];
char dept;
}
employee e1,e2;
```

4. Write a structure definition for the structure student containing examno, name and an array for storing five subject marks.

examno, name and an array for storing five subject marks.

```
struct student
{
int examno;
char name[20];
int marks[5][3];0.
}
```

5. Why for passing a structure to a function call by reference is advisable to us?

- Structures are usually passed by reference method because it saves the memory space and executes faster.

6. What is the size of the following highlighted variable in terms of byte if it is compiled in dev c++

struct A{ float f[3]; char ch[5];long double d;};
struct B{ A a; int arr[2][3];}b[3]

float f	4 Bytes
char ch	1 Byte
long double d	10 Bytes
int arr	4 Bytes

Datatype	Turbo C++	Dev C++
char	1	1
int	2	4
float	4	4
long	4	4
double	8	8
long double	10	10

7. Is the following snippet is fully correct. If not identify the error.

```
struct sum1{ int n1,n2;}s1;
struct sum2{int n1,n2}s2;
cin>>s1.n1>>s1.n2;
s2=s1;
```

```
struct sum1{ int n1,n2;}s1;
struct sum2{int n1,n2;}s2;
cin>>s1.n1>>s1.n2;
s2=s1;
```

8. Differentiate array and structure.

ARRAY	STRUCTURE
Array is a user-defined which has the sequence of data items with same data types.	Structure is a user-defined which has the combination of data items with different data types.
Syntax: data_type array_name[array_size];	Syntax: struct structure_name{data_items;}obj;
Example: int a[5];	Example: struct sum1{ int n1,n2;}s1;

9. What are the different ways to initialize the structure members?

- Values can be assigned to structure elements similar to assigning values to variables.

Example:

```
balu.rollno= “702016”;
balu.age= 18;
```

balu.weight= 48.5;

Also, values can be assigned directly as similar to assigning values to Arrays.

balu={702016, 18, 48.5};

10. What is wrong with the following C++ declarations?

A. struct point (double x, y)
B. struct point { double x, double y };
C. struct point { double x; double y }
D. struct point { double x; double y; };
E. struct point { double x; double y; }

A. struct point { double x, y; };
B. struct point { double x, double y; };
C. struct point { double x; double y; };
D. struct point { double x; double y; };
E. struct point { double x; double y; };

PART – III

Answer to all the questions (3 Marks):

1. How will you pass a structure to a function?

- A structure variable can be passed to a function in a similar way of passing any argument that is of built-in data type.
- If the structure itself is an argument, then it is called "call by value".
- If the reference of the structure is passed as an argument then it is called, "call by reference".

2. The following code sums up the total of all students name starting with 'S' and display it. Fill in the blanks with required statements.

```
struct student {int exam no,lang,eng,phy,che,mat,csc,total;char name[15];};
int main()
{
student s[20];
for(int i=0;i<20;i++)
{
Cout<<"enter the students name one by one:";
Cin>>name[i];
}
for(int i=0;i<20;i++)
{
if(name[0]=='s')
 Cout<<name[i];
}
return 0;
}
```

3. What is called nested structure. Give example

- The structure declared within another structure is called a nested structure.
- Nested structures act as members of another structure and the members of the child structure can be accessed as parent structure name.
- Child structure name. Member name.

Eg:

```
struct Student
{
```

```
int age;
float height, weight;
struct dob
{
int date;
char month[4];
int year;
};
}ob;
```

4. Re0write the following program after removing the syntactical error(s), if any. Underline each correction.

```
struct movie
{
charm_name[10];
charm_lang[10];
float ticket cost =50;};
Movie;
void main()
{
gets(m_name);
cin>>m_lang;
return 0;
}
```

```
struct movie
{
Char m_name[10];
Char m_lang[10];
float ticket_cost =50;
}Movie;
void main()
{
gets(m_name);
cin>>m_lang;
cin>>ticket_cost;
return 0;
}
```


5. What is the difference among the following two programs?

(a)
```
#include <iostream.h>
struct point { double x; double y; };
int main() {
struct point test;
test.x = .25; test.y = .75;
cout<<test.x<<test.y;
return 0;
}
```

(b)
```
#include <iostream.h>
struct { double x; double y; } Point;
int main(void) {
Point test={.25,.75};
return 0;
}
```

wrong declaration structure name should be important (syntax error will occur)

6. How to access members of a structure? Give example.

- Data members are accessed by **dot(.) operator**.

Syntax: objectname.datamember;

The student can be referred as reference name to the above structure and the elements can be accessed like student.rollno, student.age and student.weight .

7. Write the syntax and an example for structure.

Syntax:

```
struct structure_name {
type member_name1;
type member_name2;
} reference_name;
```

Example:

```
struct Student
{
long rollno;
int age;
float weight;
} ;
```

- An optional field reference_name can be used to declare objects of the structure type directly.

8. For the following structure definition write the user defined function to accept data through keyboard.

```
struct date{ int dd,mm,yy};
struct item { int itemid;char name[10];float price;date date_manif;};
```

solution:

```
date={2,55,2019};
item={109,'suji',500,'2-5-19'};
```

9. What is called anonymous structure .Give an example

- A structure without a name/tag is called anonymous structure.

Ex:

```
struct
{
long rollno;
int age;
float weight;
} student;
```

- The student can be referred as reference name to the above structure and the elements can be accessed like student.rollno, student.age and student.weight

10. Write a user defined function to return the structure after accepting value through keyboard. The structure definition is as follows

```
struct Item{int item no;float price;};
```

```
void main()
{
Cout<< "enter itemno and floatprice:"<<endl;
Cin>>itemno>>floatprice;
Cout<<itemno<<floatprice;
getch();
}
```

Part – IV

Answer to all the questions (5 Marks):

1. Explain array of structures with example

A class may contain many students. So, the definition of structure for one student can also be extended to all the students. If the class has 5 students, then 5 individual structures are required. For this purpose, an array of structures can be used. An array of structures is declared in the same way as declaring an array with built-in data types like int or char.

The following program reads the details of 5 students and prints the same.

```
#include <iostream>
using namespace std;
struct Student
{
int age;
float height;
};
void main( )
{
Student std[5];
int i;
cout<< " Enter the details for 5 students"<<endl;
for(i=0;i<20;i++)
{
cout<< " Enter the details of student"<<i+1<<endl;
cout<< " Enter the age:"<<endl;
cin>>std[i].age;
cout<< "Enter the height:"<<endl;
cin>>std[i].height;
}
cout<< "The values entered for Age and height  are"<<endl;
for(i=0;i<20;i++)
cout<<"Student "<<i+1<< "\t"<<std[i].age<< "\t"<<std[i].height;
}
```

Output:

```
Enter the details for 5 students
Enter the details of student1
Enter the age:
18
Enter the height:
160.5
Enter the details of student2
Enter the age:
18
enter the height:
164.5

The values entered for Age and height are
Student 1 18 160.5
Student 2 18 164.5
```

The above program reads age , height and weight of 5 students and prints the same details. The output is shown for only two students due to space constraints.

2. Explain call by value with respect to structure.

- The structure itself is an argument, then it is called “call by value”.
- When a structure is passed as argument to a function using call by value method, any change made to the contents of the structure variable inside the function to which it is passed do not affect the structure variable used as an argument.

Example:

```
#include <iostream>
using namespace std;
struct Employee
{
char name[50];
int age;
float salary;
};
void printData(Employee);
int main()
{
Employee p;
cout<< "Enter Full name: ";
cin>>p.name;
cout<< "Enter age: ";
cin>>p.age;
cout<< "Enter salary: ";
cin>>p.salary;
printData(p);
return 0;
}
void printData(Employee q)
{
cout<< "\n\n Displaying Information." <<endl;
cout<< "Name: " << q.name <<endl;
cout<<"Age: " <<q.age<<endl;
cout<< "Salary: " <<q.salary;
}
```

Output:

```
Enter Full name: VijayKumar
Enter age: 29
Enter salary: 34233.4

Displaying Information.
Name: VijayKumar
Age: 29
Salary: 34233.4
```

In the above example, a structure named Employee is declared and used. The values that are entered into the structure are name, age and salary of a Employee are displayed using a function named printData(). The argument for the above function is the structure Employee. The input can be received through a function named readData().

3. How call by reference is used to pass structure to a function .Give an Example

- The reference of the structure is passed as an argument then it is called, “call by reference”.
- In this method of passing the structures to functions ,the address of a structure variable /object is passed to the function using address of(&) operator. So any change made to the contents of structure variable inside the function are reflected back to the calling function.

Structures are usually passed by reference method because it saves the memory space and executes faster.

Example:

```
#include <iostream>
using namespace std;
struct Employee
{
char name[50];
int age;
float salary;
};
void readData(Employee &);
void printData(Employee);
int main()
{
Employee p;
readData(p);
printData(p);
return 0;
}
void readData(Employee &p) {
cout<< "Enter Full name: ";
cin.get(p.name, 50);
cout<< "Enter age: ";
cin>>p.age;
cout<< "Enter salary: ";
cin>>p.salary;
}
void printData(Employee p)
{
cout<< "\n\n  Displaying Information." <<endl;
cout<< "Name: " << p.name <<endl;
cout<<"Age: " <<p.age<<endl;
cout<< "Salary: " <<p.salary;
}
```

Output:

```
Enter Full name: Vijay
Enter age: 29
Enter salary: 34233.4

Displaying Information.
Name: Vijay
Age: 29
Salary: 34233.4
```

4. Write a C++ program to add two distances using the following structure definition

```
struct Distance{
int feet;
float inch;
}d1 , d2, sum;
```

```
#include <iostream>
using namespace std;
struct Distance{
   int feet;
   float inch;
}d1 , d2, sum;
int main()
{
   cout << "Enter 1st distance" << endl;
   cout << "Enter feet: ";
   cin >> d1.feet;
   cout << "Enter inch: ";
   cin >> d1.inch;
   cout << "\nEnter information for 2nd distance" << endl;
   cout << "Enter feet: ";
   cin >> d2.feet;
   cout << "Enter inch: ";
   cin >> d2.inch;
   sum.feet = d1.feet+d2.feet;
   sum.inch = d1.inch+d2.inch;
   if(sum.inch > 12)
   {
      ++ sum.feet;
      sum.inch -= 12;
   }
   cout << endl << "Sum of distances = " << sum.feet << " feet  " << sum.inch << " inches";
   return 0;
}
```

Output:

```
Enter 1st distance
Enter feet: 6
Enter inch: 3.4

Enter information for 2nd distance
Enter feet: 5
Enter inch: 10.2

Sum of distances = 12 feet  1.6 inches
```

In this program, a structure Distance containing two data members (*inch* and *feet*) is declared to store the distance in inch-feet system.

5. Write a C++ Program to Add two Complex Numbers by Passing Structure to a Function for the following structure definition

```
struct complex
{
float real;
float imag;
};
```

The prototype of the function is complex add Complex Numbers (complex, complex);

```
#include <stdio.h>
typedef struct complex
{
   float real;
   float imag;
} complex;
complex add(complex n1,complex n2);

int main()
{
   complex n1, n2, temp;
   printf("For 1st complex number \n");
   printf("Enter real and imaginary part respectively:\n");
   scanf("%f %f", &n1.real, &n1.imag);
   printf("\nFor 2nd complex number \n");
   printf("Enter real and imaginary part respectively:\n");
   scanf("%f %f", &n2.real, &n2.imag);
   temp = add(n1, n2);
   printf("Sum = %.1f + %.1fi", temp.real, temp.imag);
   return 0;
}
complex add(complex n1, complex n2)
{
     complex temp;
     temp.real = n1.real + n2.real;
     temp.imag = n1.imag + n2.imag;
     return(temp);
}
```

Output

```
For 1st complex number
Enter real and imaginary part respectively: 2.3
4.5

For 2nd complex number
Enter real and imaginary part respectively: 3.4
5
Sum = 5.7 + 9.5i
```

In this program, structures *n1* and *n2* are passed as an argument of function add().

6. Write a C++ Program to declare a structure book containing name and author as character array of 20 elements each and price as integer. Declare an array of book. Accept the name, author, price detail for each book. Define a user defined function to display the book details and calculate the total price. Return total price to the calling function.

```
#include<iostream.h>
#include<stdio.h>
#include<conio.h>
class BOOK
{
        char BOOKTITLE[20];
        char AUTHOR[20];
        int PRICE;
        void TOTAL_COST(int N)
        {
                float tcost;
                tcost=PRICE*N;
                cout<<tcost;
        }
public:
        void INPUT()
        {
                cout<<"Enter Book Name: ";
                cin>>BOOKTITLE;
                cout<<"\n Enter Book AUTHOR: ";
                gets(AUTHOR);
                cout<<"\n Enter price per copy: ";
                cin>>PRICE;
        }
        void PURCHASE()
        {
                int n;
                cout<<"\n Enter number of copies to purchase: ";
                cin>>n;
                cout<<"\n Total cost is: ";
                TOTAL_COST(n);
        }
};

void main()
{
        BOOK obj;
        obj.INPUT();
        obj.PURCHASE();
        getch();
}
```

Output:

```
Enter Book Name: Vijay-CS-Guide
Enter Book AUTHOR: Vijaykumar
Enter price per copy: 120
Enter number of copies to purchase: 10
Total cost is: 1200
```

7. Write a c++ program to declare and accept an array of professors. Display the details of the department= "COMP.SCI" and the name of the professors start with 'A'. The structure "college" should contain the following members.

prof_id as integer name and Department as character array

```
#include<iostream.h>
#include<conio.h>
struct college
{
int profid;
char dept[20], name[20];
};
void main()
{
clrscr();
int i;
college p1[3];
cout<<"Enter 3 professor details:"<<endl;
 for(i=0;i<3;i++)
 {
cout<<"Enter professor id:"<<endl;
cin>>p1[i].profid;
cout<<"Enter professor Name:"<<endl;
cin>>p1[i].name;
cout<<"Enter department:" <<endl <<endl;
cin>>p1[i].dept;
}
getch();
}
```

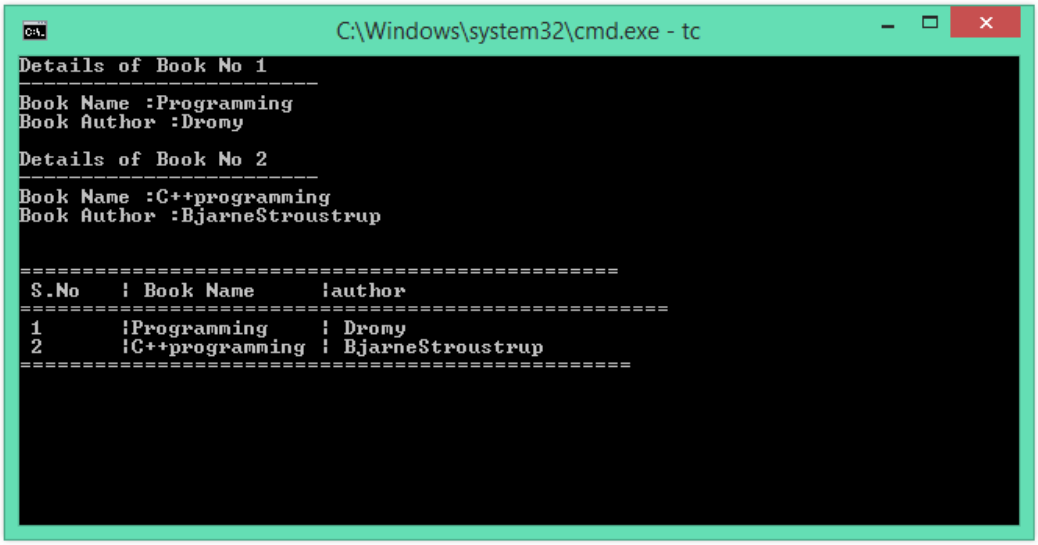
Output:

```
Enter 3 professor details:
Enter professor id: 101
Enter professor Name: Arun
Enter department: COMP.SCI

Enter professor id: 102
Enter professor Name: Vijay
Enter department: Computer Science

Enter professor id: 103
Enter professor Name: Sujitha
Enter department: Maths
```

8. Write the output of the following C++ program

```
#include<iostream>
#include<stdio>
#include <string>
#include<conio>
using namespace std;
struct books {
char name[20], author[20];
} a[50];
int main()
{
clrscr();
cout<< "Details of Book No " << 1 << "\n";
cout<< "-----------------------\n";
cout<< "Book Name :"<<strcpy(a[0].name,"Programming ")<<endl;
cout<< "Book Author :"<<strcpy(a[0].author,"Dromy")<<endl;
cout<< "\nDetails of Book No " << 2 << "\n";
cout<< "-----------------------\n";
cout<< "Book Name :"<<strcpy(a[1].name,"C++programming" )<<endl;
cout<< "Book Author :"<<strcpy(a[1].author,"BjarneStroustrup ")<<endl;
cout<<"\n\n";
cout<< "================================================\n";
cout<< " S.No\t| Book Name\t|author\n";
cout<< "================================================";
for (int i = 0; i < 2; i++) {
cout<< "\n " << i + 1 << "\t|" << a[i].name << "\t| " << a[i].author;
}
cout<< "\n================================================";
return 0;
}
```

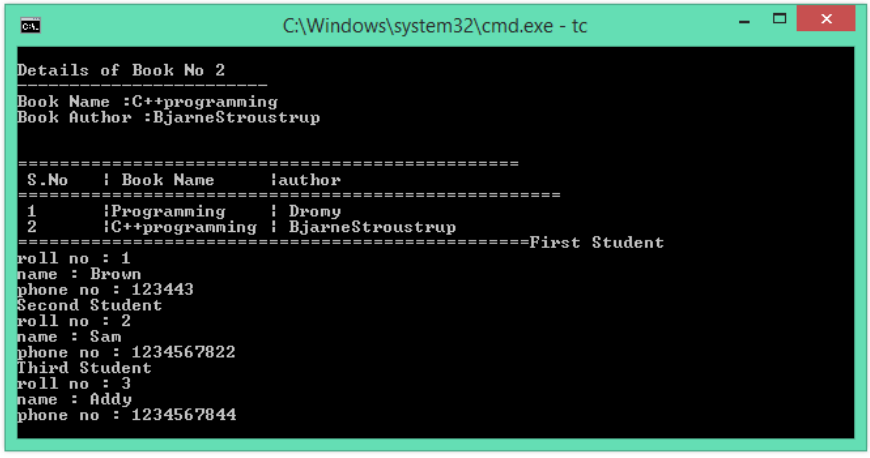
Output:



9. Write the output of the following c++ program

```
#include <iostream>
#include <string>
using namespace std;
struct student
{
introll_no;
char name[10];
long phone_number;
};
int main(){
student p1 = {1,"Brown",123443};
student p2, p3;
p2.roll_no = 2;
strcpy(p2.name ,"Sam");
p2.phone_number = 1234567822;
p3.roll_no = 3;
strcpy(p3.name,"Addy");
p3.phone_number = 1234567844;
cout<< "First Student" <<endl;
cout<< "roll no : " << p1.roll_no <<endl;
cout<< "name : " << p1.name <<endl;
cout<< "phone no : " << p1.phone_number <<endl;
cout<< "Second Student" <<endl;
cout<< "roll no : " << p2.roll_no <<endl;
cout<< "name : " << p2.name <<endl;
cout<< "phone no : " << p2.phone_number <<endl;
cout<< "Third Student" <<endl;
cout<< "roll no : " << p3.roll_no <<endl;
cout<< "name : " << p3.name <<endl;
cout<< "phone no : " << p3.phone_number <<endl;
return 0;
}
```

Output:



10. Debug the error in the following program

```
#include <istream.h>
structPersonRec
{
charlastName[10];
chaefirstName[10];
int age;
}
PersonRecPeopleArrayType[10];
voidLoadArray(PeopleRecpeop);
void main()
{
PersonRecord people;
for (i = 0; i < 10; i++)
{
cout<<people.firstName<< ‘ ‘ <<people.lastName
<<setw(10) <<people.age;
}
}
LoadArray(PersonRecpeop)
{
for (int i = 0; i < 10; i++)
{
cout<< "Enter first name: ";
cin<<peop[i].firstName;
cout<< "Enter last name: ";
cin>>peop[i].lastName;
cout<< "Enter age: ";
          cin>> people[i].age;}
```

```
#include <iostream.h>
struct PersonRec
{
char lastName[10];
char firstName[10];
int age;
} people;
void LoadArray();
void main()
{
clrscr();
PersonRec people;
int i;
for (i = 0; i < 10; i++)
{
cout<<people.firstName<< " "
<<people.lastName<<endl
<<people.age;
}
}
LoadArray(PersonRec people)
{
for (int i = 0; i < 10; i++)
{
cout<< "Enter first name: ";
cin>>people.firstName;
cout<< "Enter last name: ";
cin>>people.lastName;
cout<< "Enter age: ";
cin>> people.age;
}getch();
return 0;
}
```


M.VIJAYA KUMAR, MCA.,M.Phil.,B.Ed.,PGDCA.,
PGT-COMPUTER TEACHER,

V.SUJITHA, MCA.,M.Phil.,
PGT-COMPUTER TEACHER,

SASTRA MATRICULATION HIGHER SECONDARY SCHOOL,