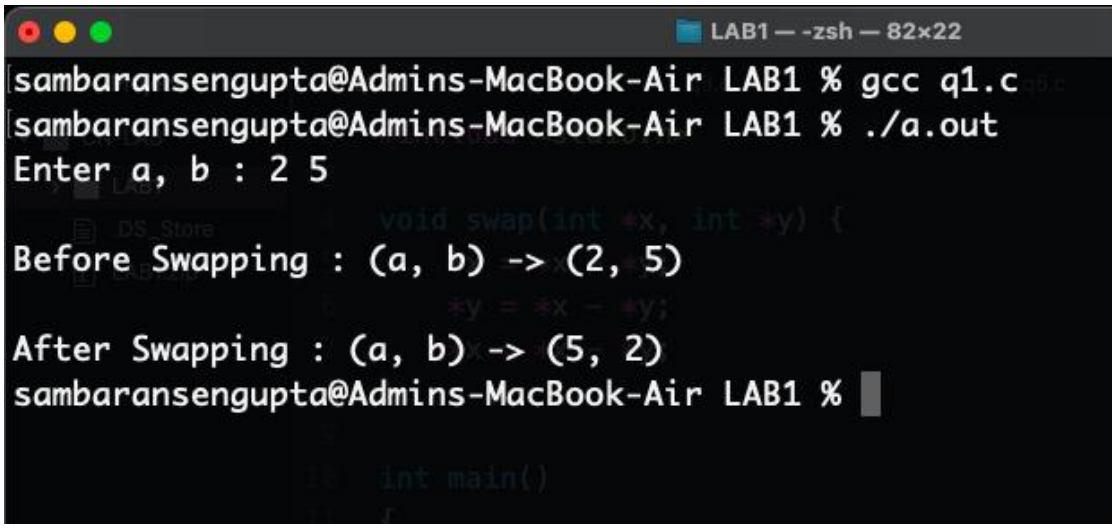
CN LAB - 01
SAMBARAN SENGUPTA
1929039

1. Write a program to swap value of two variables using pointer.

```
#include <stdio.h>
void swap(int *x, int *y) {
	*x = *x + *y;
	*y = *x - *y;
	*x = *x - *y;
}
int main()
{
	int a, b;
	printf("Enter a, b : ");
	scanf("%d%d", &a, &b);

	printf("\nBefore Swapping : ");
	printf("(a, b) -> (%d, %d)\n\n", a, b);

	swap(&a, &b);
	printf("After Swapping : ");
	printf("(a, b) -> (%d, %d)\n", a, b);

	return 0;
}
```

Output :

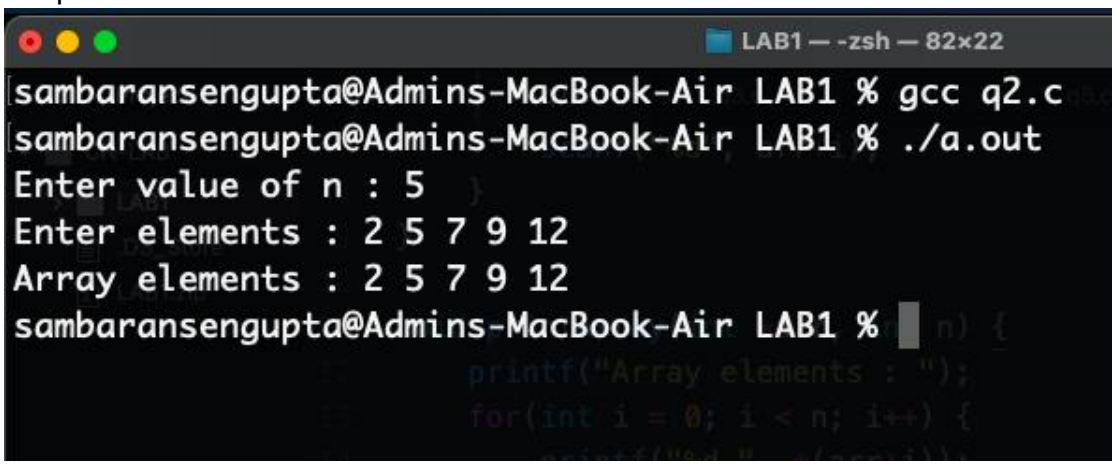


CN LAB - 01
SAMBARAN SENGUPTA
1929039

2.Write a program to input and print array elements using pointer.

```
#include <stdio.h>
#include <stdlib.h>
void input(int *arr, int n) {
	printf("Enter elements : ");
	for(int i = 0; i < n; i++) {
		scanf("%d", arr+i);
	}
}
void printArray(int *arr, int n) {
	printf("Array elements : ");
	for(int i = 0; i < n; i++) {
		printf("%d ", *(arr+i));
	}
	printf("\n");
}
int main() {
	int n;
	printf("Enter value of n : ");
	scanf("%d", &n);
	int *arr = malloc(n * sizeof(int));
	input(arr, n);
	printArray(arr, n);
	return 0;
}
```

Output :

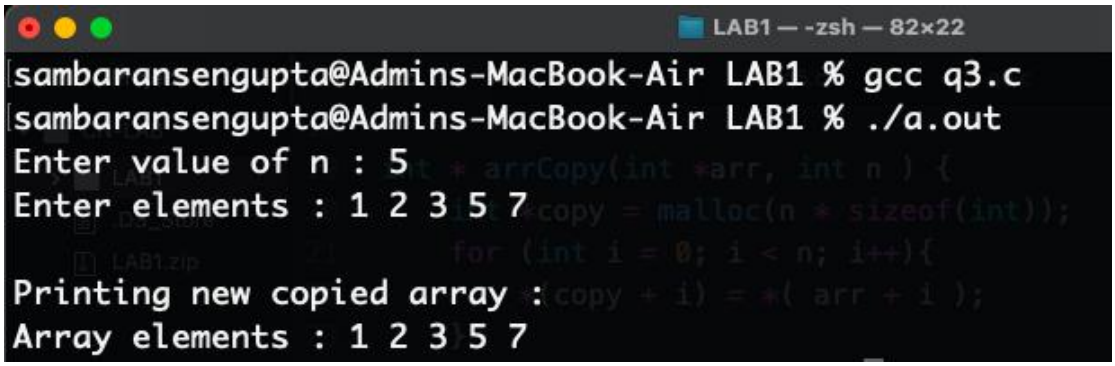


CN LAB - 01
SAMBARAN SENGUPTA
1929039

3. Write a program to copy one array to another using pointer.

```
#include <stdio.h>
#include <stdlib.h>
void input(int *arr, int n) {
	printf("Enter elements : ");
	for(int i = 0; i < n; i++) {
		scanf("%d", arr+i);
	}
}
void printArray(int *arr, int n) {
	printf("Array elements : ");
	for(int i = 0; i < n; i++) {
		printf("%d ", *(arr+i));
	}
	printf("\n");
}
int * arrCopy(int *arr, int n ) {
	int *copy = malloc(n * sizeof(int));
	for (int i = 0; i < n; i++){
		*(copy + i) = *( arr + i );
	}
	return copy;
}
int main() {
	int n;
	printf("Enter value of n : ");
	scanf("%d", &n);
	int *arr = malloc(n * sizeof(int));
	input(arr, n);
	int *copyarr = arrCopy(arr, n);
	printf("\nPrinting new copied array : \n");
	printArray(copyarr, n);
	return 0;
}
```

Output :



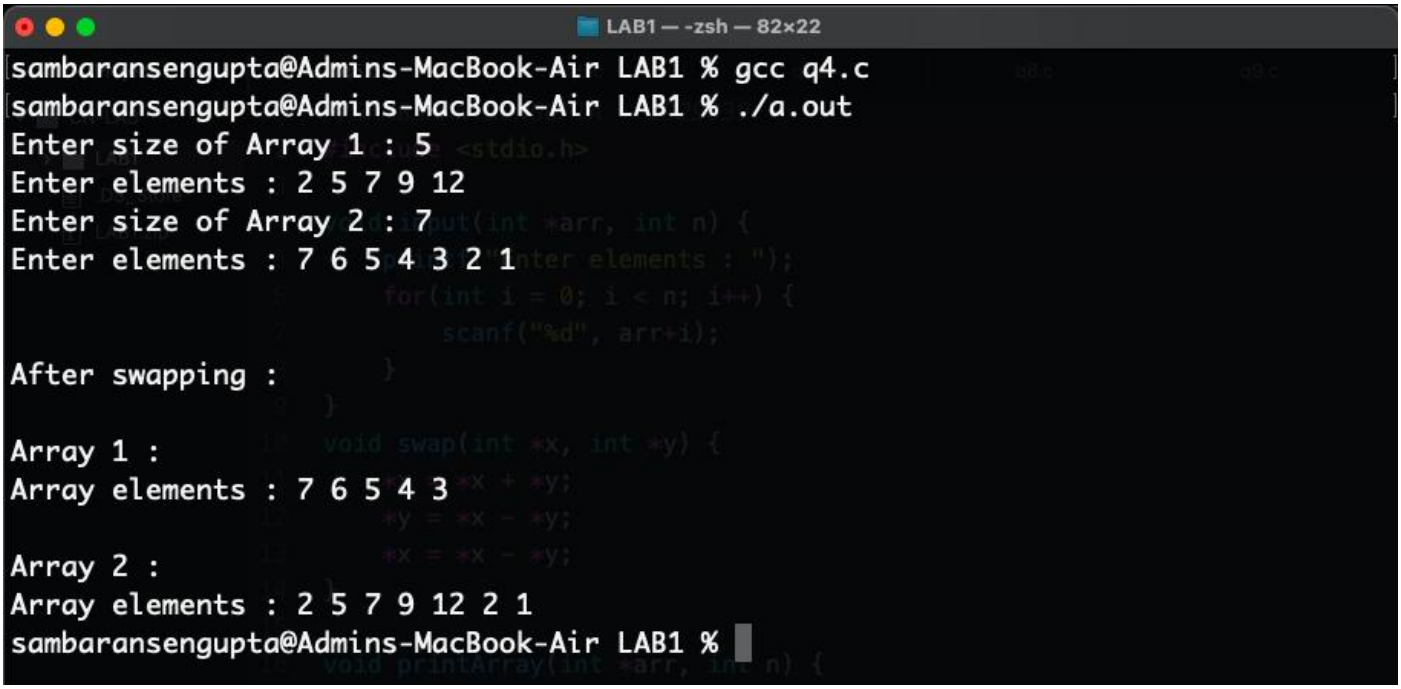
CN LAB - 01
SAMBARAN SENGUPTA
1929039

4. Write a program to swap two arrays using pointers.

```
#include <stdio.h>
#include <stdlib.h>
void input(int *arr, int n) {
	printf("Enter elements : ");
	for(int i = 0; i < n; i++) {
		scanf("%d", arr+i);
	}
}
void printArray(int *arr, int n) {
	printf("Array elements : ");
	for(int i = 0; i < n; i++) {
		printf("%d ", *(arr+i));
	}
	printf("\n");
}
void swap(int*a, int *b, int a_size, int b_size){
	int size = 0;
	if(a_size < b_size)
		size = a_size;
	else
		size = b_size;
	for(int i = 0; i < size; i++) {
		int temp = a[i];
		a[i] = b[i];
		b[i] = temp;
	}
}
int main() {
	int n, m;
	printf("Enter size of Array 1 : ");
	scanf("%d", &n);
	int arr1[n]; input(arr1, n);
	printf("Enter size of Array 2 : ");
	scanf("%d", &m);
	int arr2[m]; input(arr2, m);
	swap(arr1, arr2, n, m);
	printf("\n\nAfter swapping : \n" );
	printf("\nArray 1 : \n" );
	printArray(arr1, n);
	printf("\nArray 2 : \n" );
	printArray(arr2, m);
	return 0;
}
```

CN LAB - 01
SAMBARAN SENGUPTA
1929039

Output :



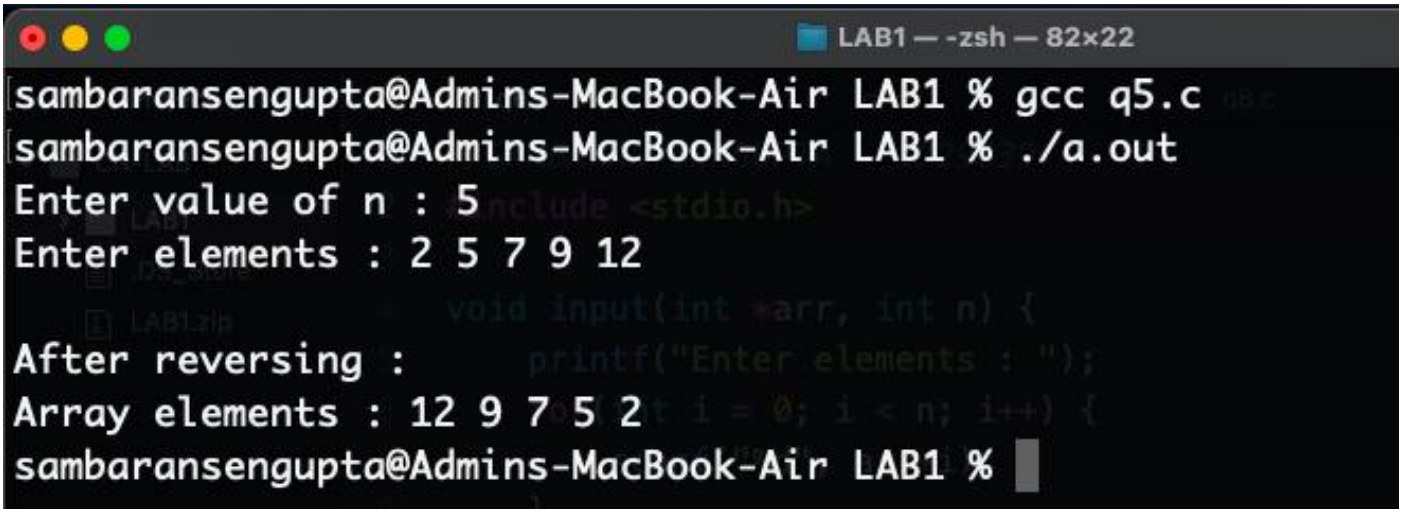

5. Write a program to reverse an array using pointers.

```
#include <stdio.h>
void input(int *arr, int n) {
	printf("Enter elements : ");
	for(int i = 0; i < n; i++) {
		scanf("%d", arr+i);
	}
}
void swap(int *x, int *y) {
	*x = *x + *y;
	*y = *x - *y;
	*x = *x - *y;
}
void printArray(int *arr, int n) {
	printf("Array elements : ");
	for(int i = 0; i < n; i++) {
		printf("%d ", *(arr+i));
	}
	printf("\n");
}
```

CN LAB - 01
SAMBARAN SENGUPTA
1929039

```
void reverse(int array[], int n)
{
	// pointer1 pointing at the beginning of the array
	int *pointer1 = array;
	// pointer2 pointing at end of the array
	int *pointer2 = array + n - 1;
	while (pointer1 < pointer2) {
		swap(pointer1, pointer2);
		pointer1++;
		pointer2--;
	}
}
int main() {
	int n;
	printf("Enter value of n : ");
	scanf("%d", &n);
	int arr[n];
	input(arr, n);
	reverse(arr, n);
	printf("\nAfter reversing : \n");
	printArray(arr, n);
	return 0;
}
```

Output :

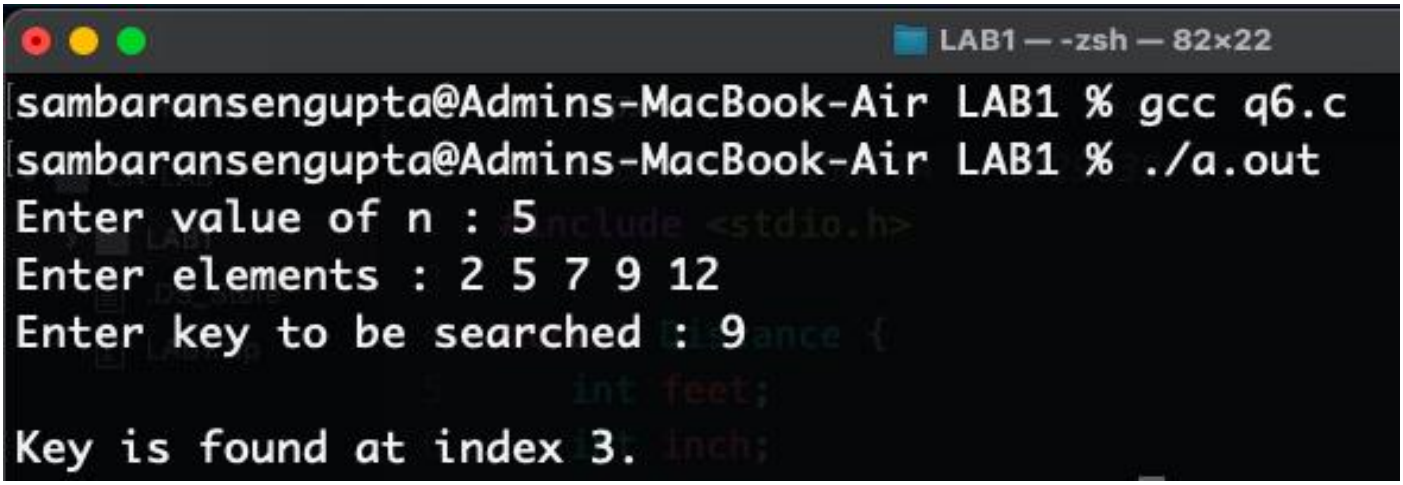


CN LAB - 01
SAMBARAN SENGUPTA
1929039

6. Write a program to search an element in array using pointers.

```
#include <stdio.h>
void input(int *arr, int n) {
	printf("Enter elements : ");
	for(int i = 0; i < n; i++) {
		scanf("%d", arr+i);
	}
}
int search(int *arr, int key, int n) {
	for(int i = 0; i < n; i++) {
		if(arr[i] == key)
			return i;
	}
	return -1;
}
int main() {
	int n;
	printf("Enter value of n : ");
	scanf("%d", &n);
	int arr[n];
	input(arr, n);

	printf("Enter key to be searched : ");
	int key;
	scanf("%d", &key);
	int index = search(arr, key, n);
	if(index == -1)
		printf("\nKey is absent in array.\n");
	else
		printf("\nKey is found at index %d.\n", index);
	return 0;
}
```

Output :



CN LAB - 01
SAMBARAN SENGUPTA
1929039

7. Write a program to add two distances in feet and inches using structure.

```
#include <stdio.h>
struct Distance {
    int feet;
    int inch;
} d1, d2, result;
int main() {
    printf("Enter 1st distance\n");
    printf("Enter feet: ");
    scanf("%d", &d1.feet);
    printf("Enter inch: ");
    scanf("%d", &d1.inch);

    printf("\nEnter 2nd distance\n");
    printf("Enter feet: ");
    scanf("%d", &d2.feet);
    printf("Enter inch: ");
    scanf("%d", &d2.inch);

    result.feet = d1.feet + d2.feet;
    result.inch = d1.inch + d2.inch;
    while (result.inch >= 12.0) {
        result.inch = result.inch - 12.0;
        ++result.feet;
    }
    printf("\nSum of distances = %d\'%d\"\n", result.feet, result.inch);
    return 0;
}
```

Output :

```
sambaransengupta@Admins-MacBook-Air LAB1 % ./a.out
Enter 1st distance
Enter feet: 2
Enter inch: 5

Enter 2nd distance
Enter feet: 3
Enter inch: 10

Sum of distances = 6'3"
```

CN LAB - 01
SAMBARAN SENGUPTA
1929039

8. Program to take marks details of 10 students and display the name of the student with highest marks.

```
#include<stdio.h>
struct student {
	int sub1, sub2, sub3, total;
	char name[50];
};
int main() {
	struct student s[10];
	int max = 0;
	struct student *topper;
	for(int i = 0; i < 10; i++) {
		printf("\nFOR STUDENT %d\n", i+1);
		printf("\nEnter Name : ");
		scanf("%s", s[i].name);
		printf("\nEnter Marks in Three Subjects = ");
		scanf("%d%d%d",& s[i].sub1,&s[i].sub2,&s[i].sub3);
		s[i].total = s[i].sub1+s[i].sub2+s[i].sub3;
		if(max < s[i].total) {
			max = s[i].total;
			topper = &s[i];
		}
	}
	printf("Student with maximum marks : %s.\n", topper->name);
	return 0;
}
```

Output :

```
FOR STUDENT 1

Enter Name : Sambaran

Enter Marks in Three Subjects = 90 20 40

FOR STUDENT 2

Enter Name : Jai

Enter Marks in Three Subjects = 40 70 80

FOR STUDENT 3

Enter Name : Rahul

Enter Marks in Three Subjects = 60 70 96
Student with maximum marks : Rahul.
sambaransengupta@Admins-MacBook-Air LAB1 %
```

CN LAB - 01
SAMBARAN SENGUPTA
1929039

9. Write a C program to compute the monthly pay of 100 employees using each employee's name, basic pay. The DA is computed as 52% of the basic pay. Gross-salary (basic pay + DA). Print the employees name and gross salary.

```
#include<stdio.h>
struct employee {
      char name[50];
      float basicpay, DA, grosspay;
}e[100];
int main() {
      int n;
      printf("Enter number of employees : ");
      scanf("%d", &n);
      for(int i = 0; i < n; i++) {
            printf("\nEnter name : ");
scanf("%s", e[i].name);
            printf("Enter Basic pay :");
            scanf("%f", &e[i].basicpay);
      }
      for(int i = 0; i < n; i++) {
            e[i].DA= 52.0 / 100 * e[i].basicpay;
            e[i].grosspay = e[i].DA + e[i].basicpay;
            printf("\n\nEmployee Name = %s \nGross Pay = %f", e[i].name, e[i].grosspay);
      }
  }
```

Output :

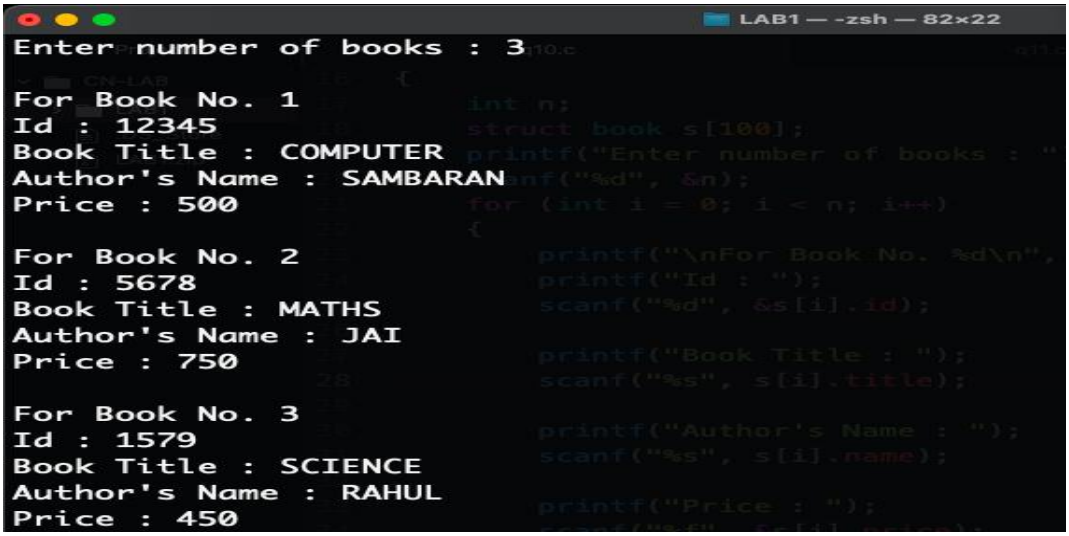
```
Enter number of employees : 3

Enter name : Sambaran
Enter Basic pay :500

Enter name : Jai
Enter Basic pay :750

Enter name : Rahul
Enter Basic pay :200


Employee Name = Sambaran
Gross Pay = 760.000000

Employee Name = Jai
Gross Pay = 1140.000000

Employee Name = Rahul
Gross Pay = 304.000000
```

CN LAB - 01
SAMBARAN SENGUPTA
1929039

10.Create a Book structure containing book_id, title, author name and price. Write a C program to pass a structure as a function argument and print the book details.

```
#include <stdio.h>
struct book {
	int id;
	char title[30];
	char name[30];
	float price;
};
void printAll(struct book s1[], int n) {
	for (int i = 0; i < n; i++)
		printf("\n\nID = %d\nTITLE = %s\nNAME = %s\nPrice = %f\n", s1[i].id, s1[i].title, s1[i].name,
s1[i].price);
}
int main() {
	int n;
	struct book s[100];
	printf("Enter number of books : ");
	scanf("%d", &n);
	for (int i = 0; i < n; i++) {
		printf("\nFor Book No. %d\n", i+1);
		printf("Id : "); scanf("%d", &s[i].id);
		printf("Book Title : "); scanf("%s", s[i].title);
		printf("Author's Name : "); scanf("%s", s[i].name);
		printf("Price : "); scanf("%f", &s[i].price);
	}
	printAll(s, n);
	printf("\n");
}
```

Output :



CN LAB - 01
SAMBARAN SENGUPTA
1929039

```
ID = 12345
TITLE = COMPUTER
NAME = SAMBARAN
Price = 500.000000


ID = 5678
TITLE = MATHS
NAME = JAI
Price = 750.000000


ID = 1579
TITLE = SCIENCE
NAME = RAHUL
Price = 450.000000
```

11. Write a program to store student name and branch using pointers to structures.

```
#include <stdio.h>
#include <stdlib.h>
struct student {
	char name[20];
	char branch[50];
};
void addStudent(struct student *s, int student_number) {
	printf("\nEnter Name : ");
	 scanf("%s", (s + student_number)->name);
	printf("Enter Branch : ");
	scanf("%s", (s + student_number)->branch);
}
void printAll(struct student *s, int size) {
	for (int i = 0; i < size; i++)
		printf("\n\nNAME = %s\nBranch = %s\n", (s+i)->name, (s+i)->branch);
}
int main() {
	printf("Enter number of students : ");
	int size;
	scanf("%d", &size);
	struct student *s = malloc(size * sizeof(struct student));
	printf("\nEnter Student Information\n");
	for(int i = 0; i < size; i++) {
		addStudent(s, i);
	}
	printAll(s, size);
}
```

CN LAB - 01
SAMBARAN SENGUPTA
1929039

Output :