

1(1)

```
#include <stdio.h>
struct person {
    char name[20];
    int age;
    float salary;
};
int main()
{
    struct person p;

    strcpy(p.name, "Golam Kibria");
    p.age = 20;
    p.salary = 2000;

    printf("Name = %s\n", p.name);
    printf("Age = %d\n", p.age);
    printf("Salary = %.0f\n", p.salary);
}
```

Output:
Name = Golam
Kibria
Age = 20
Salary = 2000

1(2)

```
#include <stdio.h>
typedef struct {
    char name[20];
    int age;
    float salary;
}Person;
int main()
{
    Person p;

    printf("Name? ");
    scanf(" %[^\\n]", p.name);
    printf("Age? ");
    scanf("%d", &p.age);
    printf("Salary? ");
    scanf("%f", &p.salary);

    printf("Name = %s\n", p.name);
    printf("Age = %d\n", p.age);
    printf("Salary = %.0f\n", p.salary);
}
```

Output:
Name ? Golam kibria
Age ? 20
Salary ? 2000
Name = Golam kibria
Age = 20
Salary = 2000

2

```
#include <stdio.h>
typedef struct {
    int age;
    float salray;
}Person;
int main()
{
    Person person1 = { 20, 2000 };
    Person person2 = { 25, 3000 };
    Person person3 = person2;

    printf("Person1 : \n");
    printf("Age = %d\n", person1.age);
    printf("Salary = %.0f\n", person1.salray);

    printf("Person2 : \n");
    printf("Age = %d\n", person2.age);
    printf("Salary = %.0f\n", person2.salray);

    printf("Person3 : \n");
    printf("Age = %d\n", person3.age);
    printf("Salary = %.0f\n", person3.salray);
}
```

Output:
Person1:
Age = 20
Salary = 2000
Person2 :
Age = 25
Salary = 3000
Person3 :
Age = 25
Salary = 3000

3

```
#include <stdio.h>
typedef struct {
    int age;
    float salray;
}Person;
int main()
{
    Person person1 = { 20, 2000 };
    Person person2 = { 25, 3000 };
    Person person3 = person2;

    if (person1.age == person2.age && person1.salray == person2.salray)
        printf("Person one is equal to person two\n");
    else
        printf("Person one is not equal to person two\n");
}
```

Output: Person one is not equal to person two

4 & 5

```
#include <stdio.h>
typedef struct {
    char name[30];
    int age;
    float salary;
}Person;
int main()
{
    Person p[10];
    int n;
    printf("How many? ");
    scanf("%d", &n);
    for (int i = 0; i < n; i++) {
        printf("Enter information for person %d\n", i + 1);
        printf("Name? ");
        scanf(" %[^\\n]", p[i].name);
        printf("Age? ");
        scanf("%d", &p[i].age);
        printf("Salary? ");
        scanf("%f", &p[i].salary);
    }
    printf("\\n");
    for (int i = 0; i < n; i++) {
        printf("Name = %s\\n", p[i].name);
        printf("Age = %d\\n", p[i].age);
        printf("Salary = %.0f\\n", p[i].salary);
    }
}
```

Output:

```
How many ? 2
Enter information for person 1
Name ? Golam Kibria
Age ? 20
Salary ? 2000
Enter information for person 2
Name ? Tausif Ahmed
Age ? 25
Salary ? 3000

Name = Golam Kibria
Age = 20
Salary = 2000
Name = Tausif Ahmed
Age = 25
Salary = 3000
```

6

```
#include <stdio.h>
typedef struct {
    char name[30];
    int age;
    float salary;
}Person;
int main()
{
    void display(Person p);

    Person person1, person2;

    strcpy(person1.name, "Golam kibria");
    person1.age = 20;
    person1.salary = 2000;
    display(person1);

    strcpy(person2.name, "Tausif Ahmed");
    person2.age = 25;
    person2.salary = 3000;
    display(person2);
}
void display(Person p)
{
    printf("Name = %s\n", p.name);
    printf("Age = %d\n", p.age);
    printf("Salary = %.0f\n", p.salary);
}
```

Output:
Name = Golam kibria
Age = 20
Salary = 2000
Name = Tausif Ahmed
Age = 25
Salary = 3000

10(1)

```
#include <stdio.h>
typedef struct {
    char ch;
}Letter;
int main()
{
    Letter l;
    l.ch = 'A';

    printf("Character is = %c\n", l.ch);
}
```

Output:
Character is = A

10(2)

```
#include <stdio.h>
typedef struct {
    int price;
    char name[30];
}Book;
int main()
{
    Book b;

    b.price = 300;
    strcpy(b.name, "Deyal");

    printf("Book name : %s\n", b.name);
    printf("Book price : %d\n", b.price);
}
```

Output:
Book name : Deyal
Book price : 300

6.3

```
#include <stdio.h>
typedef struct {
    int id;
    char name[30];
}Student;
int main()
{
    Student s;

    s.id = 2281;
    strcpy(s.name, "Kibria");
    /*Character array এর মধ্যে assignment operation
    চালানো যায় না তাই strcpy ব্যবহার করেছি*/

    printf("Student ID : %d\n", s.id);
    printf("Student Name: %s\n", s.name);
}
```

Output:
Student ID : 2281
Student Name : Kibria

6.2

```
#include <stdio.h>
typedef struct {
    int id;
    char *name;
}Student;
int main()
{
    Student s;

    s.id = 2281;
    s.name = "Kibria";
    /*Character array এর মধ্যে assignment operation
    চালানো যায় না তাই ক্যারেক্টার এর বদলে পয়েন্টার ব্যবহার করেছি*/

    printf("Student ID : %d\n", s.id);
    printf("Student Name: %s\n", s.name);
}
```

Output:

Student ID : 2281
Student Name : Kibria

6.4

/*6.3 এর মতো সেম। এটায় ইউজার হতে ইনপুট নেয়া হয়েছে*/

```
#include <stdio.h>
typedef struct {
    int id;
    char name[30];
}Student;
int main()
{
    Student s;

    printf("ID? ");
    scanf("%d", &s.id);
    printf("Name? ");
    scanf(" %[^\\n]", s.name);
    /*এইভাবে scan করাটাই উত্তম। %s দিয়ে করলে নামের দ্বিতীয় আংশটা(kibria) বাদ পরে যেত*/

    printf("Student ID : %d\\n", s.id);
    printf("Student Name: %s\\n", s.name);
}
```

Output:
ID ? 2281
Name ? Golam kibria
Student ID : 2281
Student Name : Golam kibria

6.5

```
#include <stdio.h>
#include <string.h>
typedef struct {
    char first_name[20];
    char last_name[20];
}Name;
typedef struct {
    int id;
    Name;
}Id;
int main()
{
    Id a;

    scanf("%d", &a.id);
    scanf("%s", a.first_name);
    scanf("%s", a.last_name);

    printf("ID = %d\\n", a.id);
    printf("Name = %s %s\\n", a.first_name, a.last_name);
}
```

Output:
2281
Golam
Kibria
ID = 2281
Name = Golam Kibria

6.6

```
#include <stdio.h>
#include <string.h>
typedef struct {
    char first_name[20];
    char last_name[20];
}Name;
typedef struct {
    int id;
    Name;
}Id;
int main()
{
    Id student[5];

    int n;
    printf("How many students? ");
    scanf("%d", &n);

    for (int i = 0; i < n; i++) {
        printf("Enter the ID of student %d : ", i + 1);
        scanf("%d", &student[i].id);
        printf("First Name of student %d : ", i + 1);
        scanf("%s", student[i].first_name);
        printf("Last Name of student %d : ", i + 1);
        scanf("%s", student[i].last_name);
    }
    printf("\n");
    for (int i = 0; i < n; i++) {
        printf("Information of student %d\n", i + 1);
        printf("ID : %d\n", student[i].id);
        printf("Name : %s %s\n", student[i].first_name, student[i].last_name);
    }
}
```

Output:

```
How many students ? 2
Enter the ID of student 1 : 2281
First Name of student 1 : Golam
Last Name of student 1 : kibria
Enter the ID of student 2 : 2282
First Name of student 2 : Tausif
Last Name of student 2 : Ahmed

Information of student 1
ID : 2281
Name : Golam kibria
Information of student 2
```


ID : 2282
Name : Tausif Ahmed

6.8

/*6.9 পারার জন্য এই প্রোগ্রাম টা বুঝতে হবে*/

```
#include <stdio.h>

int main()
{
    void plus1(int a);
    void plus2(int* a);

    int a = 5;
    plus1(a);
    printf("a = %d\n", a);
    plus2(&a);
    printf("a = %d\n", a);
}

void plus1(int a)
{
    a = a + 10;
}

void plus2(int* a)
{
    *a = *a + 10;
}
```

Output:

a = 5
a = 15

অর্থাৎ প্রথম ফাংশন এর ভেতরে ভেরিয়েবলের সঙ্গে 10 যোগ করলেও ফাংশনটি রিটার্ন করার সঙ্গে সঙ্গে সেই ভেরিয়েবল হারিয়ে গেছে আর মেইন ফাংশন এর এর মানের কোনো পরিবর্তন হয়নি কিন্তু দ্বিতীয় ক্ষেত্রে আমরা a-এর এড্রেস পাঠিয়ে দিয়েছি। আর সেটা গ্রহণ করেছি একটি পয়েন্টার দিয়ে। এখন সেই পয়েন্টার যাকে পয়েন্ট করে, তার যেকোনো পরিবর্তন করলেও সেটি হারিয়ে যাবে না। যেহেতু মেমোরি একেবার সঠিক জায়গায় পরিবর্তন হয়ে গিয়েছে তাই মেইন ফাংশন এর ভেতরে আসার পরেও সেটি অক্ষুণ্ণ থাকবে।

6.9

```
#include <stdio.h>
#include <string.h>
typedef struct {
    char first_name[20];
    char last_name[20];
}Name;
typedef struct {
    int id;
    Name;
    char grade[3];
}Id;
int main()
{
    void grade(Id* s, int marks);
    Id student[5];

    int n = 5, i;
    int marks[5] = { 72, 82, 69, 20, 50 };

    for (i = 0; i < n; i++) {
        printf("Enter the ID of student %d : ", i + 1);
        scanf("%d", &student[i].id);
        printf("First Name of student %d : ", i + 1);
        scanf("%s", student[i].first_name);
        printf("Last Name of student %d : ", i + 1);
        scanf("%s", student[i].last_name);
        strcpy(student[i].grade, "");
    }

    for (i = 0; i < n; i++) {
        grade(&student[i], marks[i]);
    }

    printf("\n");
    for (i = 0; i < n; i++) {
        printf("Information of student %d\n", i + 1);
        printf("ID : %d\n", student[i].id);
        printf("Name : %s %s\n", student[i].first_name, student[i].last_name);
        printf("Grade : %s\n", student[i].grade);
    }
}
```

/*ফাংশনের ভেতর সরাসরি ভ্যারিয়েবল না পাঠিয়ে, ওই ফাংশনের ঠিকানায় পাঠানো হয় তাহলে সেই ঠিকানায় ভেরিয়েবল এর কোনো পরিবর্তন করলে আমরা ফাংশনের বাইরে গেলেও সেই ভেরিয়েবল অক্ষত থাকবে। এটিকে বলে Call by reference. যা এই প্রোগ্রামটিতে ব্যবহার করা হয়েছে।*/

```
void grade(Id* s, int marks)
{
    if (marks >= 80) strcpy(s->grade, "A+");
    else if (marks >= 70) strcpy(s->grade, "A");
    else if (marks >= 60) strcpy(s->grade, "A-");
    else if (marks >= 50) strcpy(s->grade, "B");
    else if (marks >= 40) strcpy(s->grade, "C");
    else strcpy(s->grade, "F");
}
```

/*ইউজার হতে ইনপুট নিয়ে এই প্রোগ্রামটা নিজে নিজে চেষ্টা করলেই পারা যাবে ইনশাআল্লাহ্*/

Output:

```
Enter the ID of student 1 : 11
First Name of student 1 : g
Last Name of student 1 : k
Enter the ID of student 2 : 12
First Name of student 2 : t
Last Name of student 2 : a
Enter the ID of student 3 : 13
First Name of student 3 : s
Last Name of student 3 : i
Enter the ID of student 4 : 14
First Name of student 4 : a
Last Name of student 4 : r
Enter the ID of student 5 : 15
First Name of student 5 : n
Last Name of student 5 : i
```

```
Information of student 1
ID : 11
Name : g k
Grade : A
Information of student 2
ID : 12
Name : t a
Grade : A +
Information of student 3
ID : 13
Name : s i
Grade : A -
Information of student 4
ID : 14
Name : a r
Grade : F
Information of student 5
ID : 15
Name : n i
Grade : B
```

