14. Why addition of pointers are impossible ?

pointers refer to the address of a variable . adding two pointers means adding two memory address which makes no meaning and not useful so only addition of two pointers are impossible;

15. output :

```
Value of x is : 456

Value of *p1 is : 456

Value of *p2 is : 456
```

22.

```
struct ID_card
{
    int roll_no;
    char class[20];
    char name[20];
    int phone_number;
    char address[20];

}b1,b2;
```

varum create tha katu irrukanga

30. enumerate function ?

enumerate is a datatype in c

enumerate keyword is enum

enumerate is like structure

```
ex : enum st_id{
      int roll_no;
      char name[20];
      int phone_no;
    }
```

(out of syllabus)

PART C

1. Marksheet

```
#include <stdio.h>

struct mark_sheet {
    int rollno;
    char name[50];
    int english_mark;
    int tamil_mark;
    int chemistry_mark;
    int sst_mark;
    int cse_mark;
    int maths_mark;
    int result;
};

void getting_data(struct mark_sheet m) {
    printf("Enter the name of the student: ");
    scanf("%s", m.name);
    printf("Enter the roll no: ");
    scanf("%d", &m.rollno);
    printf("Enter the chemistry mark: ");
    scanf("%d", &m.chemistry_mark);
    printf("Enter the english mark: ");
    scanf("%d", &m.english_mark);
    printf("Enter the tamil mark: ");
    scanf("%d", &m.tamil_mark);
    printf("Enter the sst mark: ");
    scanf("%d", &m.sst_mark);
    printf("Enter the cse mark: ");
    scanf("%d", &m.cse_mark);
    printf("Enter the maths mark: ");
    scanf("%d", &m.maths_mark);

m.result=m.chemistry_mark+m.cse_mark+m.english_mark+m.maths_mark+m.nam
e+m.sst_mark+m.tamil_mark;
}


void printing_data(struct mark_sheet m) {
    printf("\nMarksheet : \n");
    printf("Name: %s\n", m.name);
    printf("Roll No: %d\n", m.rollno);
    printf("Chemistry Mark: %d\n", m.chemistry_mark);
    printf("English Mark: %d\n", m.english_mark);
    printf("Tamil Mark: %d\n", m.tamil_mark);
    printf("SST Mark: %d\n", m.sst_mark);
    printf("CSE Mark: %d\n", m.cse_mark);
    printf("Maths Mark: %d\n", m.maths_mark);
    printf("total : %d",m.result);
}


int main() {
    struct mark_sheet mark;
    getting_data(&mark);
    printing_data(&mark);


    return 0;
}
```

2. Union sum

```
#include<stdio.h>
union course{
        char course_name[50];
        int no_of_students;
        int duration;
    };
union college{
    char college_code[2];
    char college_name[50];
    int year_of_establishment;
    int no_courses;
    union  course courses[50];
};
void getting_data(struct college *c){
    struct college my_clg=*c;
    printf("enter the college_code : ");
    scanf("%s",&my_clg.college_code);
    printf("enter the college name : ");
    scanf("%s",&my_clg.college_name);
    printf("enter the year of est : ");
    scanf("%d",&my_clg.year_of_establishment);
    scanf("enter the no of courses : ");
    scanf("%d",&my_clg.no_courses);
    for(int i=0;i<my_clg.no_courses;i++){
        printf("enter the course name : ");
        scanf("%s",&my_clg.courses->course_name[i]);
        printf("enter the duration : ");
        scanf("%d",my_clg.courses->duration);
        scanf("enter the no of student : ");
        scanf("%d",my_clg.courses->no_of_students);
    }
}
void printing_data(struct college *c){
    struct college my_clg=*c;
    printf("college name : %s",my_clg.college_name);
    printf("college code : %d",my_clg.college_code);
    printf("year of est : %d",my_clg.year_of_establishment);
    printf("no of courses : %d",my_clg.no_courses);
    for(int i=0;i<my_clg.no_courses;i++){
        printf("course name : ",my_clg.courses->course_name[i]);
        printf("no of student : ",my_clg.courses->no_of_students);
        printf("duration : ",my_clg.courses->duration)
    }
}
int main(){
    union college my_clg;
    getting_data(&my_clg);
    printing_data(&my_clg);
}
```

3. feet-inch

```
#include<stdio.h>
struct add_two_distance_feet{
    float distance1;
    float distance2;
    float result;
};
int main(){
    struct add_two_distance_feet add;
    printf("enter the distance 1 : ");
    scanf("%f",&add.distance1);
    printf("enter the distance 2 : ");
    scanf("%f",&add.distance2);
    add.result=add.distance1+add.distance2;
    printf("sum of the distance is %.0f feets %d
inchs",add.result,(int)((add.result-(int)add.result)*100));
}
```

PART B

3. interchange two values using call by reference

```
#include<stdio.h>

void exchange_values(int *p,int *q){
    int temp;
    temp=*p;
    *p=*q;
    *q=temp;
}
int main(){
    int a=10,b=20;
    printf("values before swapping \n a : %d \t b : %d\n",a,b);
    exchange_values(&a,&b);
    printf("values after swapping \n a : %d \t b : %d",a,b);
}
```

4. what are pointer

pointer epadi assign panuva

pointer declaration

pointer 2 example with output

main_content : yes in c we can subtract two pointers. basically pointers are memory address of variables when we subtract the pointers we will get the diff btw the memory address which makes no sense . so subtraction of pointers are not prefers to be used while coding

2 examples

7. swapping two values using call by reference

```
#include<stdio.h>
```

```
void exchange_values(int *p,int *q){
    int temp;
    temp=*p;
    *p=*q;
    *q=temp;
}
int main(){
    int a=10,b=20;
    printf("values before swapping \n a : %d \t b : %d\n",a,b);
    exchange_values(&a,&b);
    printf("values after swapping \n a : %d \t b : %d",a,b);
}
```

```
8. #include <stdio.h>

int main() {
    int num1 = 10, num2 = 20, num3 = 30;
    int *ptr_array[3]; // Array of three pointers to int

    // Assigning addresses of variables to the array elements
    ptr_array[0] = &num1;
    ptr_array[1] = &num2;
    ptr_array[2] = &num3;

    // Accessing and printing values using pointers
    printf("Value at index 0: %d\n", *ptr_array[0]);
    printf("Value at index 1: %d\n", *ptr_array[1]);
    printf("Value at index 2: %d\n", *ptr_array[2]);

    return 0;
}
```

11. owm

```
12.#include<stdio.h>
struct id
{
    int roll_no;
    char name[50];
    int phone_no;
    struct address
    {
        char street_name;
        int door_no;
        char area;
    }ad;
    char class;
};
```