→ Array of pointers

```
#include<stdio.h>
 main ()
{ int marks [] = { 10, 20, 30};
    int * point [3];
    int i;
   for (int i = 0; i < 3; i++)
     { printf ("%d", marks[i];
       point [i] = & marks [i];
     }
   for ( int i = 0 ; i < 3; i++)
   { printf ("%d", *point [i]);
   }
 return (0);
 }
```

→ pointer to an array

```
int * p = & ar [0];
int * p = ar;
   p++;
```

```
include<stdio.h>
 main ()
{ int * p;
  int ar [5], i, total = 0;
  printf ("enter 5 elements");
    for (i = 0; i < 5; i++)
     { Scanf ("%d", & ar [i]);
     }
      p = ar;
```

```
printf("elements are");
for(i=0; i<5; i++)
 { printf("%d", *p);
    total = total + *p;
      p++;
 }
printf("total=%d", total);
    getch();
}
```

→ pointer to structure

```
Struct Student
{ char name[10];
  int roll;
};
Struct Student Stu;
  Struct Student *p;
    p = &Stu;
    p -> roll = 10;
    (*p).roll = 10;
```

```
#include<stdio.h>
Struct Student
{ char name[10];
  int roll;
};
int main()
{ Struct Student Stu;
  Struct Student *p;
    p = &Stu;
printf("enter name");
Scanf("%s", p->name);
printf("enter roll no");
Scanf("%d", &p->roll);
```

```
printf("roll is %d", (*p).roll);
printf("name is %s", (*p).name);
  getch();
return(0);
}
```

→ Difference between Atomic and non-atomic types

Atomic types	non-atomic types
(1.) Values of atomic data types cannot be sub divided further.	(1.) Non-atomic data types are composed of atomic types.
(2.) Atomic types can be either primitive or derived like Strings, integers, float, char etc.	(2.) Non-atomic data types are:- structures, class, union, lists etc.
(3.) Explain int, float, char in detail in 3rd point.	(3.) Explain Structures, union etc. in detail

→ if this Question is of 10 marks, then explain this 3rd point in detail, otherwise explain this 3rd point in brief.