```
//  atoi

#include <stdio.h>
#include <stdlib.h>
int main()
{
	char str[20]="121";
	int n;

	n= atoi(str);

	printf("\n N = %d", n);
	n++;
	printf("\n N = %d", n);

}
---

//  atoi

#include <stdio.h>
#include <stdlib.h>
int main()
{
	char str[20]="21PG";
	int n;

	n= atoi(str);

	printf("\n N = %d", n);
	n++;
	printf("\n N = %d", n);

}

----

//  atol

#include <stdio.h>
#include <stdlib.h>
int main()
{
	char str[20]="21PG";
	long int n;

	n= atol(str);

	printf("\n N = %d", n);
```

```
	n++;
	printf("\n N = %d", n);

}

----
//  atof

#include <stdio.h>
#include <stdlib.h>
int main()
{
	char str[20]="21.345";
	float n;

	n= atof(str);

	printf("\n N = %.3f", n);
	n=n+10;
	printf("\n N = %.3f", n);

}
----

//  atof

#include <stdio.h>
#include <stdlib.h>
int main()
{
	char str[20]="21.345";
	double n;

	n= atof(str);

	printf("\n N = %.3lf", n);
	n=n+10;
	printf("\n N = %.3lf", n);

}
----------
//  itoa

#include <stdio.h>
#include <stdlib.h>
int main()
{
	char str[20];
	int n=65536;
```

```
	itoa( n, str, 2);  //2,8,10,16

	printf("\n Str = %s", str);

}

--
#include <stdio.h>
#include<stdlib.h>
int main()
{
int a=100;
char str[100];

itoa(a,str,2);
printf("\n Binary value:%s",str);

itoa(a,str,10);
printf("\n Decimal value:%s",str);

itoa(a,str,16);
printf("\n Hexadecimal value:%s",str);

itoa(a,str,8);
printf("\n Octal value:%s",str);

return 0;
}

----------------

//structure

#include <stdio.h>

struct det;   // declaration

struct det     // definition
{
	char name[21];    // data member
	int reg;
	char section[6];
	float marks;
}s1;

int main()
{
	struct det s2;
```

```
	printf("\n Size = %d", sizeof(det) );  // bit field

}
---
// name, section, rollno, CA of 40 students.
#include <stdio.h>

struct stu_info		// declaration of structure
{
char name[22];
char section[6];
int rollno;
int CA;
}s2;	// 20+6+4+4=34 (m/y alloted is 36: due to bit padding)

// size of structure is sum of the size of all its data members
// no memory will be alloted to data members

int main()
{
	stu_info s1;

	printf("\nSize of struct = %d", sizeof(s2));


}
----
// 1. intialization
#include <stdio.h>

struct stu_info
{
char name[20];
char section[6];
int rollno;
int CA;
}s2= {"Karan", "K21PD", 02, 24};

int main()
{
	stu_info s1= {"Ajay", "K21PD", 12, 23};

	printf("\nSize of struct = %d", sizeof(stu_info));


}
----
```

```
// 2. intialization
#include <stdio.h>
#include<string.h>

struct stu_info
{
char name[20];
char section[6];
int rollno;
int CA;
}s2= {"Karan", "K21PD", 02, 24};

int main()
{
	stu_info s1;


	strcpy(s1.name,"Ajay");  // '.' is access method
	strcpy(s1.section,"K21pd");
	s1.rollno=11;
	s1.CA=23;
   }


---


// 3. intialization
#include <stdio.h>
#include<string.h>

struct stu_info
{
char name[20];
char section[6];
int rollno;
int CA;
}s2= {"Karan", "K21PD", 02, 24};

int main()
{
	stu_info s1,s3;

	strcpy(s1.name,"Ajay");  // '.' is access method
	strcpy(s1.section,"K21pd");
	s1.rollno=11;
	s1.CA=23;

	printf("\nEnter the details of student:");
```

```
	printf("\n Enter your name: ");
	gets(s3.name);
	printf("\n Enter your Section: ");
	gets(s3.section);
	printf("\n Enter your Roll No: ");
	scanf("%d", &s3.rollno);
	printf("\n Enter your CA marks: ");
	scanf("%d", &s3.CA);

	printf("\nSize of struct = %d", sizeof(stu_info));

	printf("\n Name = %s", s1.name);
	printf("\n Section = %s", s1.section);
	printf("\n Roll NO: = %d", s1.rollno);
	printf("\n CA = %d", s1.CA);

	printf("\n\n Name = %s", s2.name);
	printf("\n Section = %s", s2.section);
	printf("\n Roll NO: = %d", s2.rollno);
	printf("\n CA = %d", s2.CA);

	printf("\n\n Name = %s", s3.name);
	printf("\n Section = %s", s3.section);
	printf("\n Roll NO: = %d", s3.rollno);
	printf("\n CA = %d", s3.CA);


}
---



// 3. intialization
#include <stdio.h>
#include<string.h>

struct stu_info
{
char name[20];
char section[6];
int rollno;
int CA;
};

int main()
{
	stu_info s1,s3;

	strcpy(s1.name,"Ajay");  // '.' is access method
```

```
	strcpy(s1.section,"K21pd");
	s1.rollno=11;
	s1.CA=23;

	s3=s1;

	printf("\n\n Name = %s", s3.name);
	printf("\n Section = %s", s3.section);
	printf("\n Roll NO: = %d", s3.rollno);
	printf("\n CA = %d", s3.CA);


}
```

```
// 3. intialization
#include <stdio.h>
#include<string.h>

struct stu_info
{
char name[20];
char section[6];
int rollno;
int CA;
};

int main()
{
	stu_info s1,s3;

	strcpy(s1.name,"Ajay");  // '.' is access method
	strcpy(s1.section,"K21pd");
	s1.rollno=11;
	s1.CA=23;

	strcpy(s3.name,s1.name);
	s3.rollno=s1.rollno;

	printf("\n\n Name = %s", s3.name);
	printf("\n Section = %s", s3.section);
	printf("\n Roll NO: = %d", s3.rollno);
	printf("\n CA = %d", s3.CA);


}
```

```
// pointer to structure
#include <stdio.h>
#include<string.h>

struct stu_info
{
char name[20];
char section[6];
int rollno;
int CA;
};

int main()
{
	stu_info s3;
	stu_info *p;

	p=&s3;

	printf("\nSize of struct = %d", sizeof(s3));  //36
	 printf("\nSize of pointer = %d", sizeof(p));

	printf("\nEnter the details of student:");
	printf("\n Enter your name: ");
	gets(s3.name);
	printf("\n Enter your Section: ");
	gets(s3.section);
	printf("\n Enter your Roll No: ");
	scanf("%d", &s3.rollno);
	printf("\n Enter your CA marks: ");
	scanf("%d", &s3.CA);


	printf("\n\n Name = %s", p->name);
	printf("\n Section = %s", (*p).section);
	printf("\n Roll NO: = %d", p->rollno);
	printf("\n CA = %d", (*p).CA);


}


---

// pointer to structure data member
#include <stdio.h>
#include<string.h>
struct stu_info
{
```

```
char name[20];
char section[6];
int rollno;
int CA;
};

int main()
{
        struct stu_info s1= {"Karan", "K21PD", 02, 24};
            char *p1;
        int *p2;

            p1=s1.name;
            p2=&s1.rollno;



       printf("\n Name = %s", p1);
       printf("\n Section = %s", s1.section);
       printf("\n Roll NO: = %d", *p2);
       printf("\n CA = %d", s1.CA);

}
```