Structure Pointer

```
#include <stdio.h>

struct student {
char name[20];
int rollno;
float cpi;
};
int main()
{
struct student *studPtr, stud1;
studPtr = &stud1;
printf("Enter Name: ");
scanf("%s", studPtr->name);
printf("Enter RollNo: ");
scanf("%d", &studPtr->rollno);
printf("Enter CPI: ");
scanf("%f", &studPtr->cpi);
printf("\nStudent Details:\n");
printf("Name: %s\n", studPtr->name);
printf("RollNo: %d", studPtr->rollno);
printf("\nCPI: %f", studPtr->cpi);
return 0;
}
```

Dynamic Memory Allocation

1) malloc()

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

int main() {
int n, i, *ptr;

printf("Enter number of elements: ");
scanf("%d", &n);

ptr = (int*) malloc(n * sizeof(int));

// if memory cannot be allocated
if(ptr == NULL) {
printf("Error! memory not allocated.");
exit(0);
}


for(i = 0; i < n; ++i) {
printf("Enter elements: ");
```

```
scanf("%d", ptr + i);
printf(" %d", *(ptr+i));
}

// deallocating the memory
free(ptr);

return 0;
}
```

2) calloc()

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

int main() {
int n, i, *ptr;

printf("Enter number of elements: ");
scanf("%d", &n);

ptr = (int*) calloc(n,sizeof(int));

// if memory cannot be allocated
if(ptr == NULL) {
printf("Error! memory not allocated.");
exit(0);
}


for(i = 0; i < n; ++i) {
printf("Enter elements: ");
scanf("%d", ptr + i);
printf(" %d", *(ptr+i));
}

// deallocating the memory
free(ptr);

return 0;
}
```

3)realloc()

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
```

```
    int i,n1,n2,*ptr;
    printf("Enter the value of n1:");
    scanf("%d",&n1);
    ptr = (int*) calloc(n1,sizeof(int));
    if(ptr==NULL)
    {
    printf("Error in Memory");
    exit(0);
    }
    for(i=0;i<n1;i++)
    {
       printf("Enter the value:\n");
      scanf("%d",ptr+i);
        printf("memory=%p\n",ptr+i);
         }
    printf("Enter the value of n2");
    scanf("%d",&n2);
    ptr = realloc(ptr, n2 * sizeof(int));
    for(i=0;i<n2;i++)
    {
        printf("Enter the value:\n");
        scanf("%d",ptr+i);
        printf("reallmemory=%p\n",ptr+i);
         }

    free(ptr);
    return 0;
}
```