CSE, 2nd Year

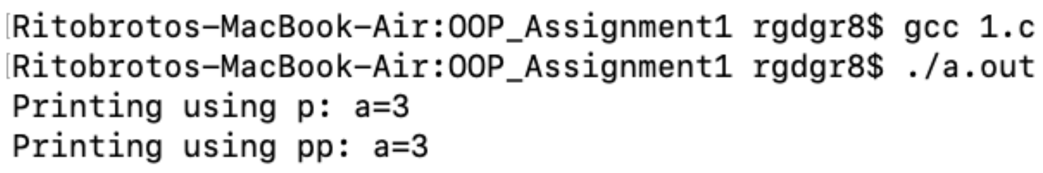
Name: RITABROTO GANGULY

ROLL: 001910501090

OOP Assignment 1

1. Write a program that will have an integer variable and a pointer (say, p) pointing to it. Also have a pointer to pointer pointing to p. Take the value for the integer variable and print it using p, and pp.

```c
#include<stdio.h>
#include<stdlib.h>
int main()
{
 int a = 3;
 int *p = &a;
 int **pp = &p;
 printf("Printing using p: a=%d\n",*p);
 printf("Printing using pp: a=%d\n",**pp);
 return 0;
}
```



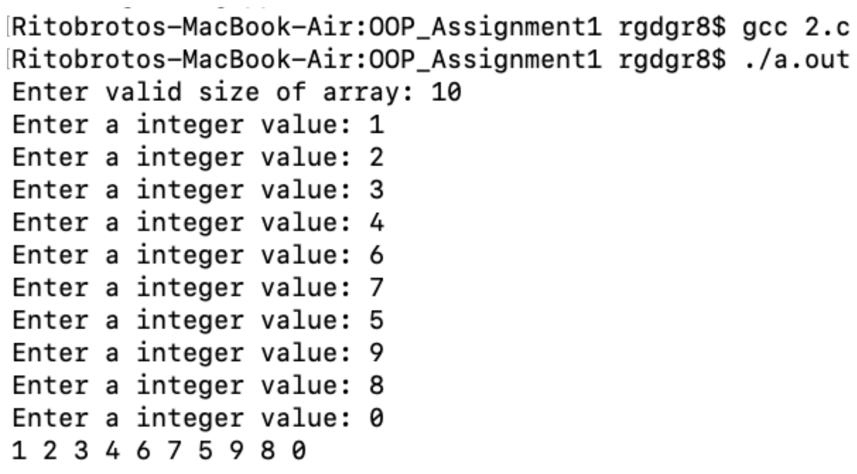
```
Ritobrotos-MacBook-Air:OOP_Assignment1 rgdgr8$ gcc 1.c
Ritobrotos-MacBook-Air:OOP_Assignment1 rgdgr8$ ./a.out
Printing using p: a=3
Printing using pp: a=3
```

2. Implement a one dimensional array of integers where array size of the array will be provided during runtime. Accept the value for the elements and print those using pointers.

```c
#include<stdio.h>
#include<stdlib.h>
#include<limits.h>
int main()
{
 int size;
 do{
   printf("Enter valid size of array: ");
   scanf("%d",&size);
 }while(size<0 && size>INT_MAX);

 int *arr = (int*)malloc(size*sizeof(int));
 for(int i=0;i<size;i++){
    printf("Enter a integer value: ");
    scanf("%d",(arr+i));
 }

 for(int i=0;i<size;i++)
     printf("%d ",*(arr+i));
 printf("\n");
 return 0;
}
```



```
Ritobrotos-MacBook-Air:OOP_Assignment1 rgdgr8$ gcc 2.c
Ritobrotos-MacBook-Air:OOP_Assignment1 rgdgr8$ ./a.out
Enter valid size of array: 10
Enter a integer value: 1
Enter a integer value: 2
Enter a integer value: 3
Enter a integer value: 4
Enter a integer value: 6
Enter a integer value: 7
Enter a integer value: 5
Enter a integer value: 9
Enter a integer value: 8
Enter a integer value: 0
1 2 3 4 6 7 5 9 8 0
```

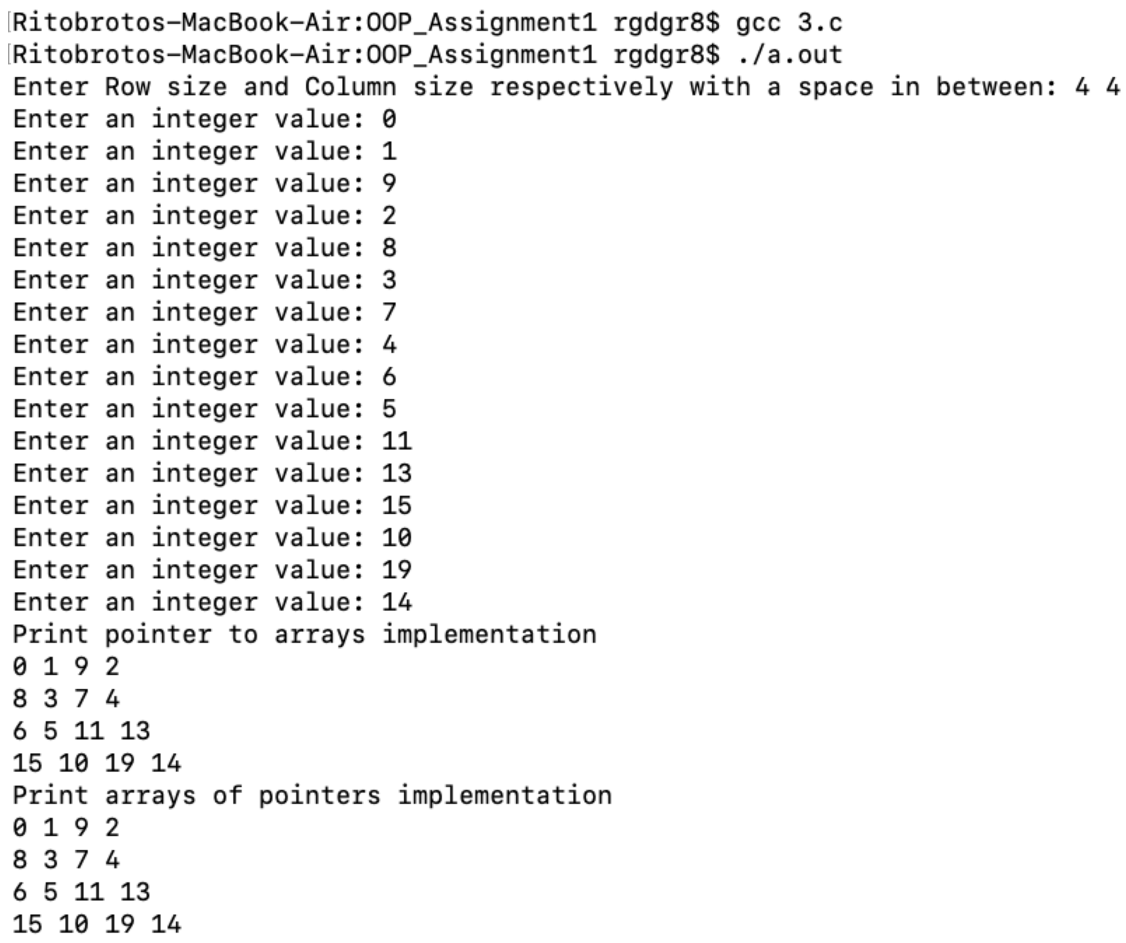
3. Implement a two dimensional array of integers using a) pointer to an array and b) array of pointers. Accept the value for the elements and print those.

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
int main()
{
 int r,c;
 printf("Enter Row size and Column size
respectively with a space in between: ");
 scanf("%d %d",&r,&c);
 int (*a)[c]; //Pointer to Array
Implementation.
 a=malloc(r*c*sizeof(int));

 int *p[r];//Array of Pointers
Implementation.
 for(int i=0;i<r;i++)
    p[i] = malloc(c*sizeof(int));

for(int j=0;j<r;j++)
 for(int i=0;i<c;i++){
    printf("Enter an integer value: ");
    int x;
    scanf("%d",&x);
    *(*(a+j)+i) = x;//input inside pointer
to arrays
    *(*(p+j)+i) = x;//input inside arrays of
pointers
 }
printf("Print pointer to arrays
implementation\n");
```

```
    for(int j=0;j<r;j++){//print pointer to
    arrays implementation
     for(int i=0;i<c;i++)
        printf("%d ",*(*(a+j)+i));
     printf("\n");
    }
    return 0;}
```



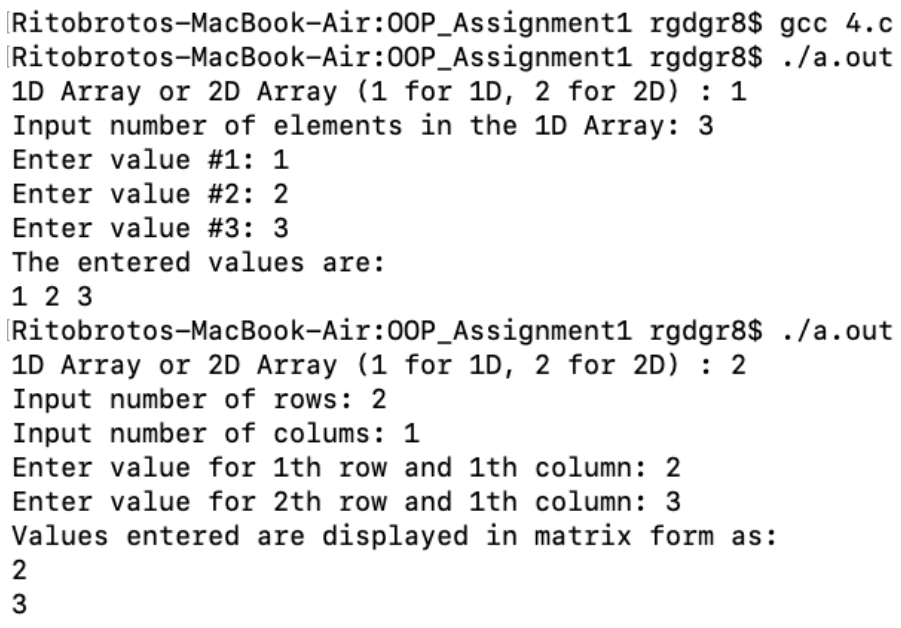
```
Ritobrotos-MacBook-Air:OOP_Assignment1 rgdgr8$ gcc 3.c
Ritobrotos-MacBook-Air:OOP_Assignment1 rgdgr8$ ./a.out
Enter Row size and Column size respectively with a space in between: 4 4
Enter an integer value: 0
Enter an integer value: 1
Enter an integer value: 9
Enter an integer value: 2
Enter an integer value: 8
Enter an integer value: 3
Enter an integer value: 7
Enter an integer value: 4
Enter an integer value: 6
Enter an integer value: 5
Enter an integer value: 11
Enter an integer value: 13
Enter an integer value: 15
Enter an integer value: 10
Enter an integer value: 19
Enter an integer value: 14
Print pointer to arrays implementation
0 1 9 2
8 3 7 4
6 5 11 13
15 10 19 14
Print arrays of pointers implementation
0 1 9 2
8 3 7 4
6 5 11 13
15 10 19 14
```

4. Implement the programs in Q.2 and 3 breaking it into functions for i) getting the dimensions from user, ii) dynamic memory allocation, iii) accepting the values and iv) printing the values.

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
 int x;
 do{
   printf("1D Array or 2D Array (1 for 1D, 2 
for 2D) : ");
   scanf("%d",&x);
 }while(x!=1 && x!=2);

 if(x==1){
   int c;
   printf("Input number of elements in the 
1D Array: ");
   scanf("%d",&c);
   int *arr = (int*)malloc(c*sizeof(int));
   for(int i=0;i<c;i++){
      printf("Enter value #%d: ",(i+1));
      scanf("%d",(arr+i));
   }
   printf("The entered values are:\n");
   for(int i=0;i<c;i++)
      printf("%d ",arr[i]);
   printf("\n");
 }else{
   int r,c;
   printf("Input number of rows: ");
   scanf("%d",&r);
   printf("Input number of colums: ");
   scanf("%d",&c);
   int **arr =
(int**)malloc(r*sizeof(int*));
   for(int i=0;i<r;i++)
      *(arr+i) =
(int*)malloc(c*sizeof(int));
   for(int i=0;i<r;i++)
      for(int j=0;j<c;j++){
         printf("Enter value for %dth row
and %dth column: ",(i+1),(j+1));
         scanf("%d",*(arr+i)+j);
      }
   printf("Values entered are displayed in
matrix form as:\n");
   for(int i=0;i<r;i++){
      for(int j=0;j<c;j++){
         printf("%d ",arr[i][j]);
      }
      printf("\n");
   }
 }

 return 0;
}
```



```
Ritobrotos-MacBook-Air:OOP_Assignment1 rgdgr8$ gcc 4.c
Ritobrotos-MacBook-Air:OOP_Assignment1 rgdgr8$ ./a.out
1D Array or 2D Array (1 for 1D, 2 for 2D) : 1
Input number of elements in the 1D Array: 3
Enter value #1: 1
Enter value #2: 2
Enter value #3: 3
The entered values are:
1 2 3
Ritobrotos-MacBook-Air:OOP_Assignment1 rgdgr8$ ./a.out
1D Array or 2D Array (1 for 1D, 2 for 2D) : 2
Input number of rows: 2
Input number of colums: 1
Enter value for 1th row and 1th column: 2
Enter value for 2th row and 1th column: 3
Values entered are displayed in matrix form as:
2
3
```

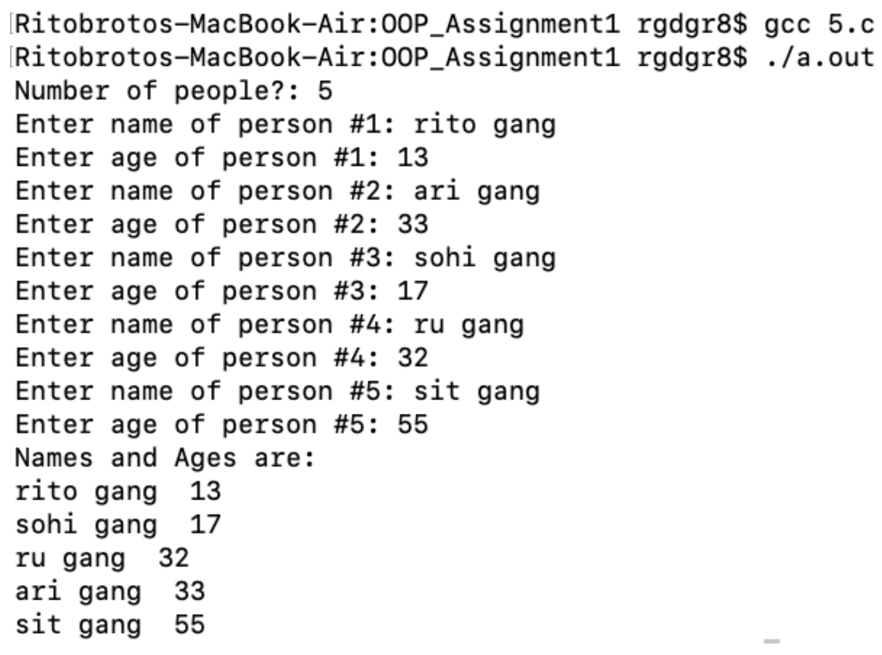
5. Store name and age of number of persons (number provided at run time). Collect the data and display data in the ascending order of age. Implement without using structure. Write functions for memory allocation of the list, sorting and display of data.

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

void sort(int ages[], char names[][100],int
p){
 for(int i=0;i<p;i++){
    for(int j=i;j<p;j++){
       if(ages[i]>ages[j]){
         int t = ages[i];
         ages[i] = ages[j];
         ages[j] = t;
         char *s = malloc(100);
         strcpy(s,names[i]);
         strcpy(names[i],names[j]);
         strcpy(names[j],s);
       }
    }
 }
}

int main()
{
 int p;
 printf("Number of people?: ");
 scanf("%d",&p);
 int ages[p];
 char names[p][100];
```

```c
    for(int i=0;i<p;i++){
       printf("Enter name of person #%d: ",
   (i+1));
       scanf(" %[^\n]%*c",names[i]);
       printf("Enter age of person #%d: ",
   (i+1));
       scanf("%d",(ages+i));
    }
    sort(ages,names,p);
    printf("Names and Ages are:\n");
    for(int i=0;i<p;i++){
       printf("%s  %d\n",names[i],ages[i]);
    }
    return 0;
   }
```



```
Ritobrotos-MacBook-Air:OOP_Assignment1 rgdgr8$ gcc 5.c
Ritobrotos-MacBook-Air:OOP_Assignment1 rgdgr8$ ./a.out
Number of people?: 5
Enter name of person #1: rito gang
Enter age of person #1: 13
Enter name of person #2: ari gang
Enter age of person #2: 33
Enter name of person #3: sohi gang
Enter age of person #3: 17
Enter name of person #4: ru gang
Enter age of person #4: 32
Enter name of person #5: sit gang
Enter age of person #5: 55
Names and Ages are:
rito gang  13
sohi gang  17
ru gang  32
ari gang  33
sit gang  55
```

6. Implement Q.5 using structure.

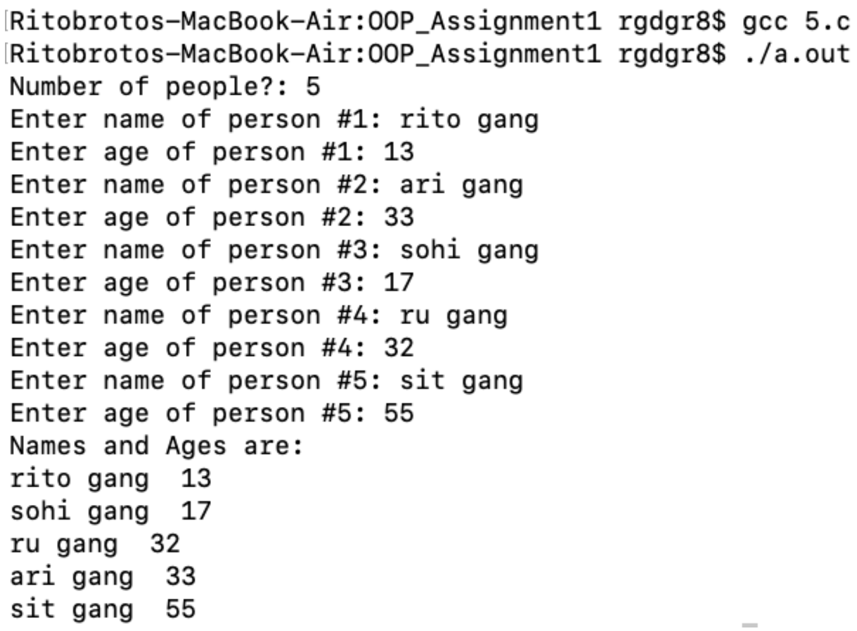
```c
#include<stdio.h>
#include<stdlib.h>
#include<string.h>

typedef struct person{
  int age;
  char name[100];
}person;

void sort(person people[],int p){
 for(int i=0;i<p;i++){
    for(int j=i;j<p;j++){
       if(people[i].age>people[j].age){
         person t = people[i];
         people[i] = people[j];
         people[j] = t;
       }
    }
 }
}

int main()
{
 int p;
 printf("Number of people?: ");
 scanf("%d",&p);
 int ages[p];
 person people[p];
```

```c
    for(int i=0;i<p;i++){
        printf("Enter name of person #%d: ",
   (i+1));
        scanf(" %[^\n]%*c",people[i].name);
        printf("Enter age of person #%d: ",
   (i+1));
        scanf("%d",(ages+i));
        people[i].age = ages[i];
    }
    sort(people,p);
    printf("Names and Ages are:\n");
    for(int i=0;i<p;i++){
        printf("%s
   %d\n",people[i].name,people[i].age);
    }
    return 0;
   }
```



```
Ritobrotos-MacBook-Air:OOP_Assignment1 rgdgr8$ gcc 5.c
Ritobrotos-MacBook-Air:OOP_Assignment1 rgdgr8$ ./a.out
Number of people?: 5
Enter name of person #1: rito gang
Enter age of person #1: 13
Enter name of person #2: ari gang
Enter age of person #2: 33
Enter name of person #3: sohi gang
Enter age of person #3: 17
Enter name of person #4: ru gang
Enter age of person #4: 32
Enter name of person #5: sit gang
Enter age of person #5: 55
Names and Ages are:
rito gang  13
sohi gang  17
ru gang  32
ari gang  33
sit gang  55
```

7. Maintain a list to store roll, name and score of students. As and when required student record may be added or deleted. Also, the list has to be displayed. Design suitable functions for different operations.

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

int nodes = 0;//number of nodes
int n;//number of subjects

typedef struct student{
 int roll;
 int* scores;
 char name[100];
}student;

typedef struct node{
 student s;
 struct node *next;
}node;

void create(node **h,node **t,student x);
void insert(node **h,node **t,student x,int
pos);
void delete(node **h,node **t,int pos);

void print_list(node *h){
 int k = 1;
 for(;h!=NULL;h=h->next){
     printf("%d)\nRoll: %d  Name: 
%s\nScores: ",k,h->s.roll,h->s.name);
     k++;
     for(int i=0;i<n;i++)
        printf("%d ",h->s.scores[i]);
     printf("\n");
 }
}

void input(student *s){
 printf("Enter a Roll Number and a Name: ");
 scanf("%d %s",&(s->roll),s->name);
 s->scores = malloc(n*sizeof(int));
 for(int i=1;i<=n;i++){
    printf("Enter a score for Subject #%d: 
",i);
    scanf("%d",&(s->scores[i-1]));
 }
}

int main()
{
 node *head,*tail;
 student s;
 printf("Enter the number of subject: ");
 scanf("%d",&n);
 input(&s);
 create(&head,&tail,s);//list created and 
first node added.
 char c;
do{
```

```c
 printf("Type d/D to delete an element, i/I
to insert an element, p/P for printing the
list: ");
// c = getchar();
 scanf(" %c",&c);
 int p;
 if(c=='d' || c=='D'){
do{
    printf("Enter a position to delete: ");
    scanf("%d",&p);
   }while(p<=0 && p>nodes);
   delete(&head,&tail,p);
 }else if(c=='i' || c=='I'){
   do{
    printf("Enter a position to insert in:
");
    scanf("%d",&p);
   }while(p<=0);
   input(&s);
   if(nodes>0)
     insert(&head,&tail,s,p);
   else
     create(&head,&tail,s);
 }else {
  print_list(head);
 }
}while(c=='i' || c=='I' || c=='p' || c=='P'
|| c=='d' || c=='D');
 return 0;
}

void create(node **h,node **t,student x){
 node *newNode = malloc(sizeof(node));
 newNode->s = x;
 newNode->next = NULL;
 *h = newNode;
 *t = *h;
 nodes++;
}
void insert(node **h,node **t,student x,int
pos){
 if(pos<1){//illegal position
   printf("No such Position!\n");
   return;
 }

 if(pos>=(nodes+1)){//append to the end of
the list
   node *newNode = malloc(sizeof(node));
   newNode->s = x;
   newNode->next = NULL;
   (*t)->next = newNode;
   *t = newNode;
   nodes++;
   return;
 }

 /*------Insert in middle or front------*/
 node *temp = *h;
 int traversed = 1;
 while(traversed<(pos-1)){
     temp = temp->next;
     traversed++;
 }
 node *newNode = malloc(sizeof(node));
 newNode->s = x;
 if(pos>1){
   newNode->next = temp->next;
```

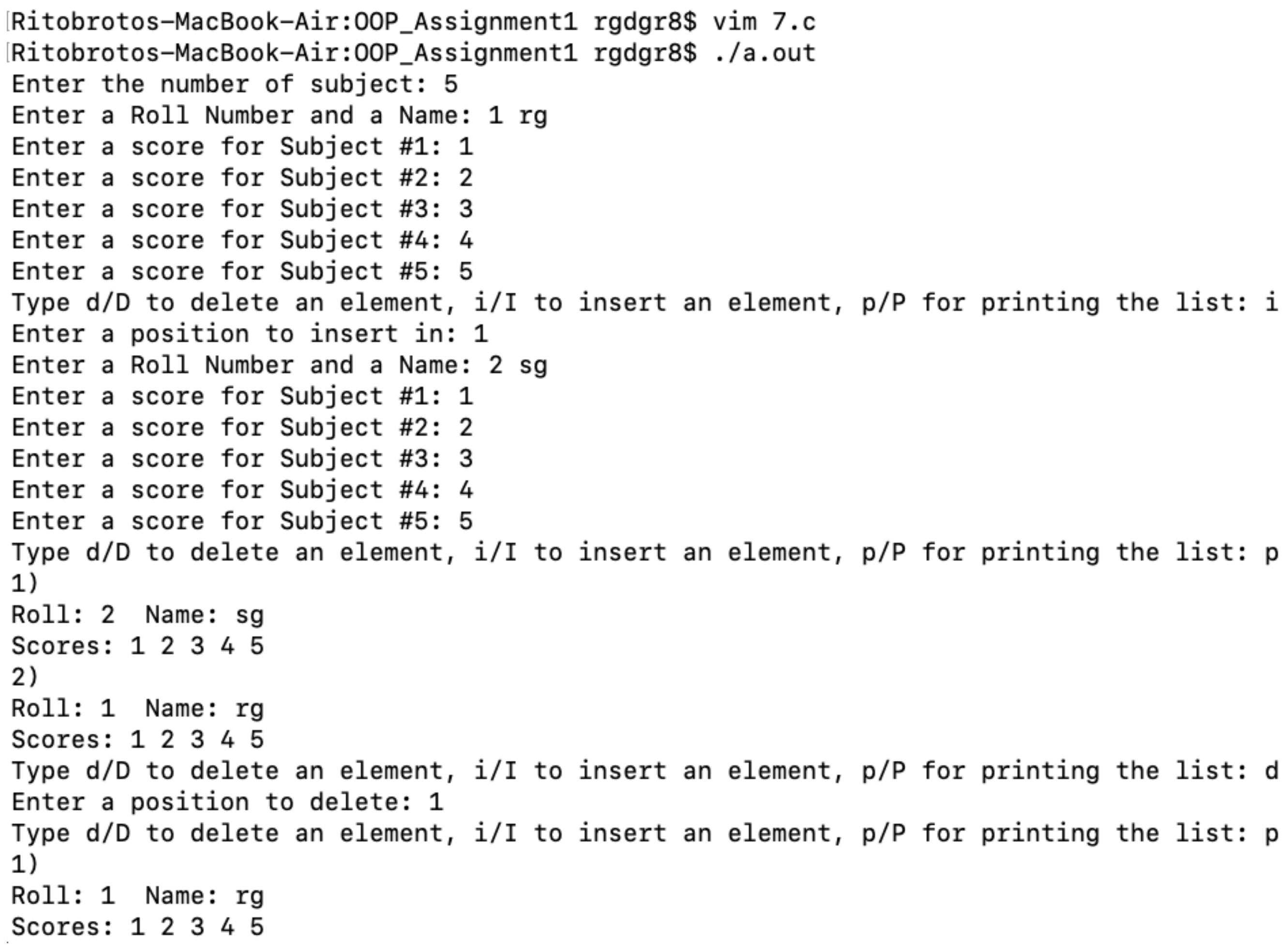
```
    temp->next = newNode;
  }
  else{
    newNode->next = *h;
    *h = newNode;
    if(nodes<1)
      *t = *h;
  }
  nodes++;
}

void delete(node **h,node **t,int pos){
  if(pos>nodes || pos<1){//illegal position
    printf("No such data!\n");
    return;
  }
  node *temp = *h;
  if(pos==1){//deleting first node
    *h = (*h)->next;
    free(temp);
    nodes--;
    return;
  }

/*-----deleting any other node than first
node-----*/
  int traversed = 1;
  while(traversed<(pos-1)){
        temp = temp->next;
        traversed++;
  }

if(temp->next->next==NULL)
     *t = temp;
  node *del = temp->next;
```

```c
     temp->next = temp->next->next;
     free(del);
     nodes--;
    }
```



```
Ritobrotos-MacBook-Air:OOP_Assignment1 rgdgr8$ vim 7.c
Ritobrotos-MacBook-Air:OOP_Assignment1 rgdgr8$ ./a.out
Enter the number of subject: 5
Enter a Roll Number and a Name: 1 rg
Enter a score for Subject #1: 1
Enter a score for Subject #2: 2
Enter a score for Subject #3: 3
Enter a score for Subject #4: 4
Enter a score for Subject #5: 5
Type d/D to delete an element, i/I to insert an element, p/P for printing the list: i
Enter a position to insert in: 1
Enter a Roll Number and a Name: 2 sg
Enter a score for Subject #1: 1
Enter a score for Subject #2: 2
Enter a score for Subject #3: 3
Enter a score for Subject #4: 4
Enter a score for Subject #5: 5
Type d/D to delete an element, i/I to insert an element, p/P for printing the list: p
1)
Roll: 2  Name: sg
Scores: 1 2 3 4 5
2)
Roll: 1  Name: rg
Scores: 1 2 3 4 5
Type d/D to delete an element, i/I to insert an element, p/P for printing the list: d
Enter a position to delete: 1
Type d/D to delete an element, i/I to insert an element, p/P for printing the list: p
1)
Roll: 1  Name: rg
Scores: 1 2 3 4 5
```