

Question 1. Compute factorial of a number using recursion

Constraint:

$0 \leq n \leq 10$

Code:

```
#include<stdio.h>
int fact(int n)
{
    // base case
    if(n==0)
        return 1;
    return n*fact(n-1);
}
int main()
{
    int n;
    scanf("%d",&n);
    int ans=fact(n);
    printf("%d\n",ans);
    return 0;
}
```

Question 2. Find the nth fibonacci number where fibonacci sequence is

0, 1, 1, 2, 3, 5, 8, 13, 21, ....

Constraint:

$0 \leq n \leq 20$

Code:

```
#include<stdio.h>
int fibo(int n)
{
    // base case
    if(n==0)
        return 0;
    if(n==1)
        return 1;
    return fibo(n-1)+fibo(n-2);
}

int main()
{
    int n;
```

```

scanf("%d",&n);
int ans=fibo(n);
printf("%d\n",ans);
return 0;
}

```

Question 3. How to hash values of array if values are negative also ?

Constraints:

$1 \leq n \leq 100000$

$-100000 \leq a[i] \leq 100000$

Code:

```

#include<stdio.h>
#define offset 100000
int arr[100022],hash_arr[200022];
int main()
{
    int n,i;
    scanf("%d",&n);
    for(i=0;i<n;i++)
        scanf("%d",&arr[i]);
    for(i=0;i<n;i++)
    {
        hash_arr[arr[i]+offset]++;
    }
    for(i=0;i<=200000;i++)
    {
        printf("%d %d\n",i-offset,hash_arr[i]);
    }
    return 0;
}

```

Time Complexity:  $O(2 \cdot 10^5)$

Question 4. Merge sort

$1 \leq n \leq 100000$

$1 \leq a[i] \leq 10^9$

Code:

```

#include<stdio.h>
int temp[100022]; //temp array for merging
void mergesort(int a[],int i,int j)
{

```

```

int mid;

if(i<j)
{
    mid=(i+j)/2;
    mergesort(a,i,mid);    //left recursion
    mergesort(a,mid+1,j);  //right recursion
    merge(a,i,mid,mid+1,j); //merging of two sorted sub-arrays
}
}

void merge(int a[],int i1,int j1,int i2,int j2)
{
    int i,j,k;
    i=i1; //beginning of the first list
    j=i2; //beginning of the second list
    k=0;
    while(i<=j1 && j<=j2) //while elements in both lists
    {
        if(a[i]<a[j])
            temp[k++]=a[i++];
        else
            temp[k++]=a[j++];
    }
    while(i<=j1) //copy remaining elements of the first list
        temp[k++]=a[i++];
    while(j<=j2) //copy remaining elements of the second list
        temp[k++]=a[j++];
    //Transfer elements from temp[] back to a[]
    for(i=i1,j=0;i<=j2;i++,j++)
        a[i]=temp[j];
}

int main()
{
    int a[100000],n,i;
    scanf("%d",&n);
    for(i=0;i<n;i++)
        scanf("%d",&a[i]);

    mergesort(a,0,n-1);
    for(i=0;i<n;i++)
        printf("%d ",a[i]);
    return 0;
}

```

Time Complexity:  $O(n \log n)$

Question 5. Sorting the array of structure.(Home work question)

$1 \leq n \leq 1000$

Code:

```
#include<stdio.h>
#include<string.h>
#define N 1000
struct data
{
    char name[100];
    int age;
}arr[N+1];
void sort_structure(data arr[],int n)
{
    int i,j;
    for(i=0;i<n;i++)
    {
        for(j=i+1;j<n;j++)
        {
            //if arr[i].name is greater
            if(strcmp(arr[i].name,arr[j].name)>0)
            {
                //here we swap the structure
                char temp[100];
                strcpy(temp,arr[i].name);
                strcpy(arr[i].name,arr[j].name);
                strcpy(arr[j].name,temp);
                int temp2;
                temp2=arr[i].age;
                arr[i].age=arr[j].age;
                arr[j].age=temp2;
            }
        }
    }
}
int main()
{
    int i,n;
    scanf("%d\n",&n);
    for(i=0;i<n;i++)
        scanf("%s %d",arr[i].name,&arr[i].age);
    sort_structure(arr,n);
    for(i=0;i<n;i++)
        printf("%s %d\n",arr[i].name,arr[i].age);
    return 0;
}
```

Time Complexity:  $O(n*n)$