Answer to the question no-01

Memoization method:

```
#include<bits/stdc++.h>
#define ll long long int
using namespace std;
const int I=1000;
ll DP[I];
ll Fibonacci(int n)
{
   if(n==0)
   {
      DP[0]=0;
      return 0;
   }
   if(n==1)
   {
      DP[1]=1;
      return 1;
   }
   if(DP[n]!=-1)
   {
      return DP[n];
   }
   int ans=Fibonacci(n-1)+Fibonacci(n-2);
   DP[n]=ans;
   return ans;
}
int main()
{
   int a;
   cin>>a;
   memset(DP,-1,sizeof(DP));
   cout<<Fibonacci(a)<<endl;
}
```

Tabulation method:

```
#include<bits/stdc++.h>
#define ll long long int
using namespace std;
const int I=1000;
ll DP[I];
int main()
{
   int a;
   cin>>a;
   DP[0]=0;
   DP[1]=1;
   for(int i=2;i<=a;i++)
   {
      DP[i]=DP[i-1]+DP[i-2];
   }
   cout<<DP[a]<<endl;
}
```

Answer to the question no-02

Memoization method:

```
#include <bits/stdc++.h>
using namespace std;
#define ll long long int
const int I = 1e5;
ll dp[I];
vector<ll>a(I);
ll Farida(int n)
{
   if(n<=0)
   {
      return 0;
   }
   if(n==1)
   {
      return a[1];
```

```
    }
    if(dp[n]!=-1)
    {
        return dp[n];
    }
    ll ans1=a[n]+Farida(n-2);
    ll ans2=a[n]+Farida(n-3);
    dp[n]=max(ans1,ans2);
    return dp[n];
}
int main()
{
    int t;
    cin>>t;
    for (int c = 1; c <= t; c++) {
        int n;
        cin>>n;
        for (int i = 1; i <= n; i++) {
            cin>>a[i];
        }
        memset(dp,-1,sizeof(dp));
        ll mx=-1;
        for(int i=n;i>=1;i--)
        {
            mx=max(mx,Farida(i));
        }
        cout<< "Case "<<c<< ": "<<mx<<endl;
    }
}
```

Tabulation method:

```
#include <bits/stdc++.h>
using namespace std;
#define ll long long int
const int I = 1e5;
ll dp[I];
int main()
```

```
{
   int t;
   cin>>t;
   for (int c = 1; c <= t; c++) {
      int n;
      cin>>n;
      vector<ll> a(n+5);
      for (int i = 1; i <= n; i++) {
         cin>>a[i];
      }
      dp[1] = a[1];
      for (int i = 2; i <= n; i++) {
         dp[i] = max(dp[i-1], i > 2 ? dp[i-2] + a[i] : a[i]);
      }
      cout<< "Case "<<c<< ": "<<dp[n]<<endl;
   }
}
```

Answer to the question no-03

Memoization method:

```
#include<bits/stdc++.h>
using namespace std;
#define ll long long int
const int I=1e5+5;
int a;
ll dp[I];
vector<ll>A;
ll F[I];
ll boredom(int n)
{
   if(n<=0)
   {
      return 0;
   }
   if(n==1)
   {
```

```
        return F[1];
    }
    if(dp[n]!=-1)
    {
        return dp[n];
    }
    ll ans=max({F[n]*n+boredom(n-2),boredom(n-1)});
    dp[n]=ans;
    return dp[n];
}
int main()
{
    cin>>a;
    memset(dp,-1,sizeof(dp));
    memset(F,0,sizeof(F));
    A.resize(a+5);
    ll mx=0;
    for(int i=1;i<=a;i++)
    {
        int b;cin>>b;
        A[i]=b;
        F[b]++;
        if(mx<b)
        {
            mx=b;
        }
    }
    cout<<boredom(mx)<<endl;

}
```

Tabulation method:

```
#include<bits/stdc++.h>
using namespace std;
#define ll long long int
const int I=1e5+5;
ll dp[I];
```

```
int main()
{
   int n;
   cin>>n;
   vector<ll>A(100005);
   for(int i=1;i<=n;i++)
   {
      int a;cin>>a;
      A[a]++;

   }
   dp[1]=A[1];
   for(int i=2;i<=100004;i++)
   {
      dp[i]=max(dp[i-1],dp[i-2]+A[i]*i);
   }
   cout<<dp[100004]<<endl;

}
```

Answer to the question no-04

Memoization method:

```
#include <bits/stdc++.h>
using namespace std;
#define ll long long int
const int I = 1e5;
ll dp[I];
ll N_tribonacci(int n)
{
   if(n==0)
   {
      return 0;
   }
   if(n<=2)
   {
      return 1;
```

```
    }
    if(dp[n]!=-1)
    {
        return dp[n];
    }
    ll ans=N_tribonacci(n-1)+N_tribonacci(n-2)+N_tribonacci(n-3);
    dp[n]=ans;
    return ans;
}
int main()
{
    int n;
    cin>>n;
    memset(dp,-1,sizeof(dp));
    cout<<N_tribonacci(n)<<endl;
}
```

Tabulation method:

```
#include <bits/stdc++.h>
using namespace std;
#define ll long long int
const int I = 1e5;
ll dp[I];
int main()
{
    int n;
    cin>>n;
    dp[0]=0;
    dp[1]=dp[2]=1;
    for(int i=3;i<=n;i++)
    {
        dp[i]=dp[i-1]+dp[i-2]+dp[i-3];
    }
    cout<<dp[n]<<endl;
}
```

Answer to the question no-05

Memoization method:

```
#include<bits/stdc++.h>
using namespace std;
#define ll long long int
const int I=1e5+5;
int dp[I];
int target_1(int n)
{
   if(n==1)
   {
      return 0;
   }
   if(dp[n]!=0)
   {
      return dp[n];
   }
   int ans=1+target_1(n-1);
   if(n%2==0)
   {
      ans=min(ans,target_1(n/2)+1);
   }
   if(n%3==0)
   {
      ans=min(ans,target_1(n/3)+1);
   }
   dp[n]=ans;
   return dp[n];
}
int main()
{
   int n;
   cin >> n;
```

```
    memset(dp,0,sizeof(dp));
    cout<<target_1(n)<<endl;
}
```

Tabulation method:

```
#include<bits/stdc++.h>
using namespace std;
#define ll long long int
const int I=1e5+5;
int dp[I];
int main()
{
    int n;
    cin >> n;
    dp[1] = 0;
    for(int i=2; i<=n; i++)
    {
        dp[i] = dp[i-1] + 1;
        if(i%2 == 0)
        {
            dp[i] = min(dp[i], dp[i/2] + 1);
        }
        if(i%3 == 0)
        {
            dp[i] = min(dp[i], dp[i/3] + 1);
        }
    }
    cout << dp[n] << endl;
}
```