\_\_\_\_\_\_

(15 points) Use the following table to find the maximum content of the sack using <u>Dynamic</u>
 <u>Programming</u>. The max weight the sack can hold is 15 lb. You May use the program to create the table.

item	1	2	3	4	5
Value(\$)	7	9	5	12	14
Weight(lb)	3	4	2	6	7

```
#include <iostream>
#include <iomanip>
#include <algorithm>
using namespace std;
// returns the maximum value that can be put in a knapsack of capacity W
int knapSack(int W, int wt[], int val[], int n) {
    int i, w;
    int K[n+1][W+1]; // K[n+1][W+1]
    // build table K[][] in bottom up manner
    for (i = 0; i <= n; i++) {
        for (w = 0; w \le W; w++) {
            if (i == 0 | | w == 0) K[i][w] = 0;
            else if (wt[i-1] \le w) K[i][w] = max(val[i-1] + K[i-1][w - wt[i-1]],
K[i-1][w]);
            else K[i][w] = K[i - 1][w];
        }
    for (int i = 0; i <= n; ++i) {
        for (int w = 0; w <= W; w++) cout << setw(5) << K[i][w];
        cout << endl;</pre>
    return K[n][W];
int main() {
    // number of items and capacity of the knapsack
    int n = 5, W = 15;
    int val[5] = { 7, 9, 5, 12, 14 }, wt[5] = { 3, 4, 2, 6, 7 };
    int maxValue = knapSack(W, wt, val, n);
```

```
cout << "\nMaximum value in the Knapsack = " << maxValue << endl;
return 0;
}</pre>
```

## WRITTEN WORK:

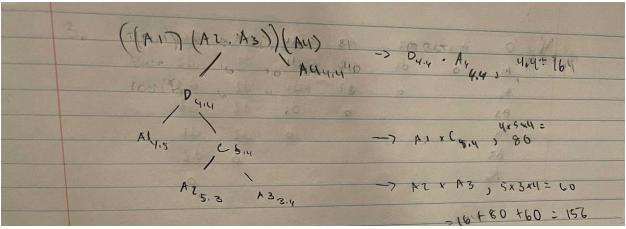
1. trem 1 1 2 13 14 1 =
Value(b) 7 9 5 12 14 1516
Weight (18) 3 4 2 6 7
0 000 7 1 0 1 7
Weight, 600 1 22 3004 5 06 2 9 0 2
1 67
bag=213 0 0 0 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
क के शिंड प्र. १ कि मा अस्तार कि मा स्ट्राइ
bag= 2 2 3 0 0 7 9 9 9 16 16 16 16 16 16 16 16
\$13
£13 £23 £13 £1,23 £1,23 £1,23
bag { 1, 6,3} 0 0 5 7 9 12 14 16 16 21 21 21 21 21 21 21 21
हु उड़े हाड़ हाड़िश्चरा है। हो है। हो है। हो हो है। है। हो है। है। हो है। है। हो है। है। हो है। है। हो है।
buy {1,2,343 0 0 5 7 9 12 14 16 17 71 21 24 26 28 BB
bag(1,,5} 0 0 5 7 9 12 14 16 17 21 24 26 28 30 33
1 14 10 18 30 33
£1, c, 3, 43
Max Valve = 33
Max Value = 33
A STATE OF THE STA

2. (15 points) Given the following chain of matrices multiplications

Matrices: A1 \* A2 \* A3 \* A4 Dimensions: 4.5 5.3 3.4 4.4

Use Dynamic Programming to determine in what order we should multiply those matrices to obtain minimum number of multiplications. Show all work for maximum credit.

1						
	2. Matrices A1 × A2 × A3 × A4					
	Dimensions! 4,5 5,3 3,4 4,4					
	LE DO LE VIENNESS					
	minimum number of multiplications					
	Land of the state					
	A14.5 A2 A3 A4 4.4					
	£ 5.3 3.4 4.4					
	do= 4 d1= 5 d2= 3 d3= 4 d4=4					
	W 21 31 31 31 31 31 31 31 31 31 31 31 31 31					
	DAIAZASAY MAIAZASAU					
	A1 0 01=5 d1=5 d3=4 A1 0 60 104 +56					
	Az 0 d2=3 d3=4 Az 0 60 108					
	143 0 d3=4 143 0 48					
1 1	Au 60 Au					
1						
134	$M(1,1) = A1.A2 = (1 \times 5 \times 3 = 60 \times M(1,1))$					
	$M(2,3) = A2.A3 = 5 \times 5 \times 4 = 60$ $M(8,4) = A3.A4 = 3 \times 4 \times 4 = 48$					
	M(i,)) = m & M (i,u) + M (u+1) + di -1 +du+d)3, ic=a=i					
	M(1,3) = min & M(1,1) + M(2,3) + 00 × 01 x d3, M(1,2) + M(3,3) 100					
	i=1 J=3					
	$\frac{150 \text{ min } 20 + 60 + 4x5x4}{20 + 0 + 4x3x4}$ $= \frac{150 \text{ min num of}}{20 + 0 + 2x5x4}$					
IME 42=	156 min num of = min & 140, 1083 = 108					
-	M(2.4) = min & M[2,2], M(3,4) + d1 x d2 x d4 , M(2,3) + m(4,4) + d1 x d3 x d4					
	= min 2 0 + 48 + 5 x 3 x 4 (0 + 0 + 5 x 4 x 4)					
	100 3 108 1003					
	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1					
	3 6 - 108 + 41 × × 41					
	= mine 188, 1963 = 156					
ALC: NO.						



3. (15 points) Use the following table to construct an OBST

Items	1	2	3	4
Data	10	20	30	40
Cost(\$)	6	4	2	3

## Code to find optimal BST:

```
#include <bits/stdc++.h>
using namespace std;
int sum(int freq[], int i, int j);
int optCost(int freq[], int i, int j)
    if (j < i)
        return 0;
    if (j == i)
        return freq[i];
    int fsum = sum(freq, i, j);
    int min = INT_MAX;
    for (int r = i; r \le j; ++r)
        int cost = optCost(freq, i, r - 1) +
                   optCost(freq, r + 1, j);
        if (cost < min)</pre>
            min = cost;
    return min + fsum;
```

```
int optimalSearchTree(int keys[],
                      int freq[], int n)
    return optCost(freq, 0, n - 1);
int sum(int freq[], int i, int j)
    int s = 0;
    for (int k = i; k \le j; k++)
    s += freq[k];
    return s;
// Driver Code
int main()
    int keys[] = {10, 20, 30, 40};
    int freq[] = {6, 4, 2, 3};
    int n = sizeof(keys) / sizeof(keys[0]);
    cout << "Cost of Optimal BST is "</pre>
         << optimalSearchTree(keys, freq, n);
    return 0;
```

## **OUTPUT:**

```
[Running] cd "c:\Users\Nick\Desktop\Algorithm-Engineering-Work\Homework 4\" &&
g++ OBST.cpp -o OBST && "c:\Users\Nick\Desktop\Algorithm-Engineering-
Work\Homework 4\"OBST
Cost of Optimal BST is 28
[Done] exited with code=0 in 0.994 seconds
```

Written Work:

ritten Wo	
21	I+cmy , 2 3 4
	Dara 10 20 30 40
	cost (3) 6 4 2 3
	ty treeso based on catal on Wumbers 3
11	010 ( 0,0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
	10 (3.7 4) (3.36)
	7 30 5000 012173 07.47-
	3 40 9) (20 1841742 + 2271 446 =1
	30 (10,0) -0 (10,13-6) (1018) 10 (10,14) -
2)	10
	10) (1xu+2x6+3x3+4x2)=
	30
Jan 1	
3)	16 30 1 1 10 10 10 10 10 10 10 10 10 10 10 1
	20 39
	1 2 1 2 1 1 2 2
	(0.20 35
4)	10 30 13) .20.3 144 24 6 + 212 + 3+3
	10 30.40 = 29
	10
	1 x 6 + 2 x 2 + 8 x 3 + 4 x 4 = 35  14)  10  10  10  10  10  10  10  10  10  1
5)	10 40
	20
	1 × 6 + 2×3 + 3×4+ 4×2 = 32
6	
6	10,40
(1)	20 11- 211
	126 + 2×3 + 3×2 + 4×4=34
7	2) 20. 14 + 2 × 6 + 3 × 2 + 4 × 3 =
	10230°40
	40

4. (10 points)Use the following table to group each chain of matrices to minimize the number of multiplications

D	A1	A2	A3	A4	A5	A6
A1	0	d1	d2	d1	d1	d2
A2		0	d2	d1	d2	d3
A3			0	d1	d2	d3
A4				0	d2	d1
A5					0	d2
A6						0

Complete the following table

Chain	Grouping with min multiplications (just write the grouping order)
A1.A2.A3.A4.A5.A6	
A2.A3.A4.A5.A6.A7	
A3.A4.A5.A6	
A4.A5.A6	

y D	
AI AZ	A3 AV A5 A6
AI o di	dz di di dz
A2 o	d2 d1 d2 d3
Ay	0 01 02 03
As Man	0 2 2
Ab	0 62
AN THERE CAN SA CT	
31 - 93 + 0 2 4/3.	
Chain	Group
AI AZ AZ AY AS AG	= (A) A) (ABAYI AS AG), D(AI, AG) = dz
81 8 813 463 451	= (A1 A) (A3 A4 A5) (A6) , D(A3, A6) = 23
AT AT AT SERVICE	= (A) AD (((A3 A4)(A5)).16) D(A3, A5) = dz
\$10 per 17 %	A LAND STATE OF THE STATE OF TH
A 2 A 3 A 4 A 5 A 6	(02 02 02) (0 02)
(A7 DNE)	= (A2 A3 A4) (A5 A6) D(A2, A6) = 03 = (A2) · (A3 A4) (A5 A6) D(R2, A4) = 01
(177702)	(1,21, (1,31,41) 1, O(1,41) 20)
A3 A4 A5 A6	= (A3 A4 A5) (A6) . D(AZ A4) = d3
of es ed	= (A3 A4 A5) (A6), D(A3, A6) = d3
MIONE	
248 4 FM) 05 (P	The state of the s
E ST IS EMPLY SEE	2 + 3 × 5 + & × 6 2
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	
AU AS AG	= (A4). (A5 A6) , D(A4, AC)= 21
JAP Y SAB STAN	of the second

5. Write a program to display the first 10 Fibonacci numbers using the Cassini's identity  $F(n-1)F(n-2) - [F(n)]^2 = (-1)^n$ , n>=1

```
#include <iostream>
using namespace std;

size_t s[100] = { 0 };
int cassini(int n) { return (n & 1) ? -1 : 1; }

size_t fib(const int& n) {
    if (s[n]) return s[n];
    if (n <= 2) return s[n] = 1;
    else
    {
        s[n] = fib(n-1) + fib(n-2);
        return s[n];
    }
}

int main(int argc, char** argv) {
    for(int i = 0; i < 11; i++){
        cout << fib(i) << " " << cassini(i) << endl;
    }
    return 0;
}</pre>
```

## **OUTPUT:**

```
[Running] cd "c:\Users\Nick\Desktop\Algorithm-Engineering-Work\Homework 4\" &&
g++ fib.cpp -o fib && "c:\Users\Nick\Desktop\Algorithm-Engineering-Work\Homework
4\"fib
1
          1
          -1
          1
2
          -1
3
          1
5
          -1
8
          1
13
           -1
21
           1
34
           -1
55
           1
[Done] exited with code=0 in 0.421 seconds
```