Name:Ram Haygrev S Roll No: 231901039

EXP 4 DYNAMIC PROGRAMMING

1)

```
#include <stdio.h>

unsigned long long countWays(int n) {
    unsigned long long dp[n + 1];


    for (int i = 0; i <= n; i++)
        dp[i] = 0;

    dp[0] = 1;

    for (int i = 1; i <= n; i++) {
        dp[i] += dp[i - 1]; // using 1
        if (i >= 3)
            dp[i] += dp[i - 3]; // using 3
    }

    return dp[n];
}

int main() {
    int n;
    scanf("%d", &n);
    if (n < 0) {
        printf("0\n");
        return 0;
    }
    printf("%llu\n", countWays(n));
    return 0;
}
```

2)

```
#include <stdio.h>
#define MAX 100
```

```
int max(int a, int b) {
    return (a > b) ? a : b;
}

int main() {
    int n, board[MAX][MAX], dp[MAX][MAX];

    scanf("%d", &n);

    for (int i = 0; i < n; i++)
        for (int j = 0; j < n; j++)
            scanf("%d", &board[i][j]);


    dp[0][0] = board[0][0];


    for (int j = 1; j < n; j++)
        dp[0][j] = dp[0][j - 1] + board[0][j];


    for (int i = 1; i < n; i++)
        dp[i][0] = dp[i - 1][0] + board[i][0];

    for (int i = 1; i < n; i++) {
        for (int j = 1; j < n; j++) {
            dp[i][j] = board[i][j] + max(dp[i - 1][j], dp[i][j - 1]);
        }
    }


    printf("%d\n", dp[n - 1][n - 1]);

    return 0;
}
```

3)

```
#include <stdio.h>
#include <string.h>
#define MAX 1000

int max(int a, int b) {
    return (a > b) ? a : b;
}

int main() {
    char s1[MAX], s2[MAX];
    scanf("%s %s", s1, s2);

    int n = strlen(s1);
    int m = strlen(s2);
    int dp[MAX][MAX];

    for (int i = 0; i <= n; i++)
        dp[i][0] = 0;
    for (int j = 0; j <= m; j++)
        dp[0][j] = 0;


    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= m; j++) {
            if (s1[i - 1] == s2[j - 1])
                dp[i][j] = 1 + dp[i - 1][j - 1];
            else
                dp[i][j] = max(dp[i - 1][j], dp[i][j - 1]);
        }
    }

    printf("%d\n", dp[n][m]);

    return 0;
}
```

4)

```
#include <stdio.h>

#define MAX 1000

int max(int a, int b) {
    return (a > b) ? a : b;
}

int main() {
    int n;
    scanf("%d", &n);

    int arr[MAX], dp[MAX];
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
        dp[i] = 1;  // Initialize DP
    }


    for (int i = 1; i < n; i++) {
        for (int j = 0; j < i; j++) {
            if (arr[j] <= arr[i]) {
                dp[i] = max(dp[i], dp[j] + 1);
            }
        }
    }


    int max_len = 0;
    for (int i = 0; i < n; i++)
        max_len = max(max_len, dp[i]);

    printf("%d\n", max_len);

    return 0;
}
```