

### Question 3:-

A polynomial of degree  $n$  (in one variable, with real coefficients) is an expression of the form:  $a_n x^n + a_{n-1} x^{n-1} + a_{n-2} x^{n-2} + \dots + a_2 x^2 + a_1 x + a_0$  where  $a_n, a_{n-1}, a_{n-2}, \dots, a_2, a_1, a_0$  are real numbers. Example:  $3x^4 - 2x^2 + 5x + 1$  is a polynomial of degree 4.

Write a complete C program that reads a polynomial of degree  $n$  through command line arguments. That is, when the program is executed as `./a.out v a_n a_{n-1} a_{n-2} ... a_2 a_1 a_0`, it uses `"int argc"` and `"char *argv[]"` to read  $a_n, a_{n-1}, a_{n-2}, \dots, a_2, a_1, a_0$  and the value of  $x$  (say,  $v$ ) for which the polynomial is to be evaluated.

The program then creates  $n+1$  child processes  $P_0, P_1, P_2, \dots, P_n$  such that  $P_i, 1 \leq i \leq n$ , evaluates the  $i$ th term of the polynomial. That is  $P_0$  evaluates  $a_n v^n$ ,  $P_1$  evaluates  $a_{n-1} v^{n-1}$ , and so on.  $P_n$ , in addition to evaluating the  $n$ th term ( $a_0$ ), also prints the value of the whole polynomial for  $v$ .

Please note that your program should be well-documented and properly indented for easy reading!

Code:-

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
// #include <sys/wait.h> --> Tried to implement without using wait()

/*
 * We have a coefficients array that stores the coefficient of the polynomial
 * in order, eg:- coefficients[0]=a_n-1, coefficients[1]=a_n-2.... coefficients[n-1]=a_0, n being the degree
 * We will use this array to evaluate the final value of polynomial for a given value v
 */

// Evaluate the value of the polynomial for a given value of v
double evaluatePolynomialTerm(int n, double v, double coefficients[])
{
    double result = coefficients[0];
    for (int i = 1; i <= n; i++)
    {
        result = result * v + coefficients[i];
    }
    return result;
}
```

```

// Print each polynomial term
void printPolynomialTerm(int index, double x, double coefficients[], int n)
{
    int res = 1;
    for (int i = n - index - 1; i >= 0; i--)
    {
        res *= x;
    }
    printf("P(%d): Result for term %d = %.0f * %d = %.0f\n", n - index, n - index, coefficients[index], res, coefficients[index] * res);
}

int main(int argc, char *argv[])
{
    double v = atof(argv[1]); // Value of v
    int n = argc - 3;        // Number of terms
    double totalSum = 0;
    double coefficients[n + 1];
    for (int i = 0; i <= n; i++)
    {
        coefficients[i] = atof(argv[i + 2]);
    }
    printf("Details of Polynomial Chosen: \nDegree: %d\nValue of X: %.0f\n\n", n, v);

    for (int i = 0; i <= n; i++)
    {
        pid_t pid = fork();

        if (pid == 0)
        {
            printPolynomialTerm(i, v, coefficients, n);
            exit(0);
        }
        else
        {
            // wait(NULL);
            sleep(n * 0.2);
            if (i == n)
            {
                totalSum = evaluatePolynomialTerm(n, v, coefficients);
                printf("Total sum of the polynomial: %.0f\n", totalSum);
            }
        }
    }
    exit(0);
}

```

