Program to calculate age by finding difference between current date and date of birth.

Presented by:

- IEC2021063 Javesh Lodha
- IEC2021064 Yash Gupta
- IEC2021065 Shashi Kumar Chaubey
- IEC2021066 Pavitra Pandey

Under the supervision of:
Dr. Mohammad Javed

3rd February 2022

Abstract

In this paper we have devised an algorithm to calculate age by finding the difference between the current date and the date of birth.

Algorithm Design

- We have defined a structure named “date” here which contains three int data types named as days, months and years respectively.
- Now, after storing both the dates, we will take the difference between the current date and the date of birth.
- If the difference between the two dates is less than 0, we will carry over one month and add 30.
- Since every month does not have 30 days, we will add a correction factor over the months to equalize it.
- Now, we will continue to subtract the months of the two dates.

- If the difference is smaller than 0, we will borrow one from the year and add 12 to the difference of months.

- Continuing this, we will calculate the difference between the years. If the difference between the two is negative, this means that the current date is before the date of birth, which is invaild, so we will output an invalid warning.

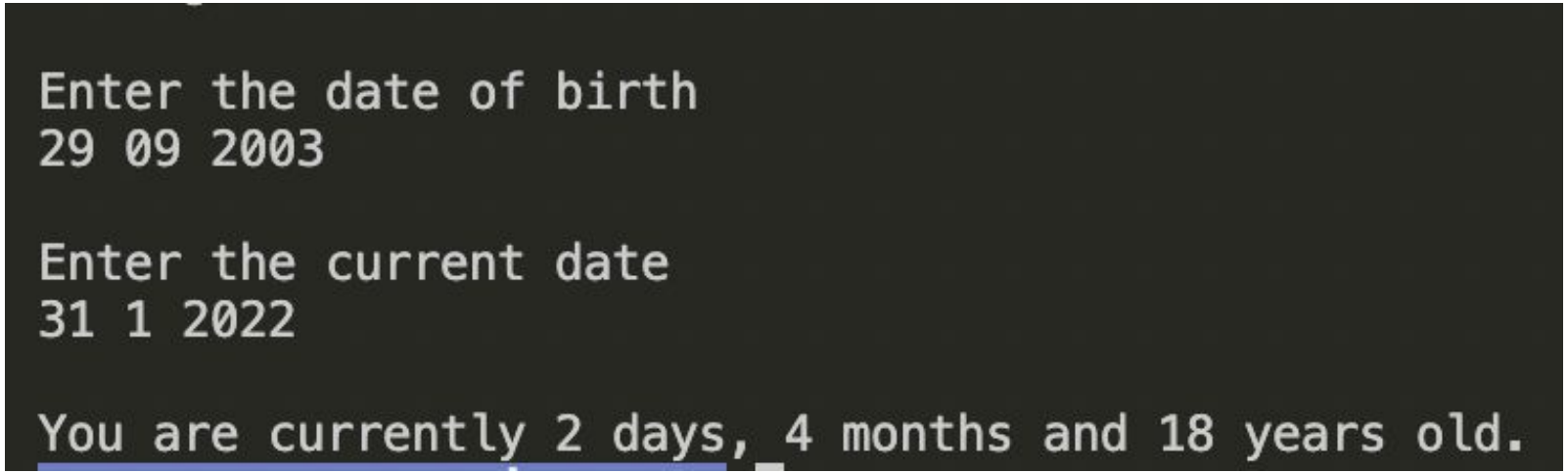
- Now, we will display the difference between the two dates in the form of days, months and years.

This is the code for the problem.

Made in c

```c
#include <stdio.h>

struct date
{
    int days;
    int months;
    int years;
};

int main()
{
    struct date dob;
    struct date current;
    printf("\nEnter the date of birth\n");
    scanf("%d%d%d", &dob.days, &dob.months, &dob.years);

    printf("\nEnter the current date\n");
    scanf("%d%d%d", &current.days, &current.months, &current.years);

    struct date dif;

    dif.days = current.days - dob.days;
    int correction = 0;
    if (dif.days < 0)-
    {
        dif.days += 30;
        current.months--;
        switch (current.months)
        {
        case 1:
        case 3:
        case 5:
        case 7:
        case 8:
        case 10:
        case 12:
            correction = 1;
            break;
        case 2:
            correction = -2;
            break;
        }
    }

    dif.days += correction;

    dif.months = current.months - dob.months;
    if (dif.months < 0)
    {
        dif.months += 12;
        current.years--;
    }

    dif.years = current.years - dob.years;
    if (dif.years < 0)
    {
        printf("\nWrong format between the current and the date of birth");
        return 0;
    }
    printf("\nYou are currently %d days, %d months and %d years old.", dif.days, dif.months,
dif.years);
}
```

Input / output



Conclusion

So, we have used the properties of structures in this code to reduce the complexity of the code and make it more presentable while dealing with heterogeneous data types.

References

- https://www.geeksforgeeks.org/structures-c/
- https://stackoverflow.com

The end