DATA STRUCTURE AND PROGRAMMING I

Enumeration

```
enum{
            YES,
            NO,
            NoResponse
};
```

```
enum Status{
            DELIVERED=1,
            PENDING=2,
            CANCELED=3,
            SUCCESS=4
};
enum Status STATUS;
```

```
enum Season{
            Summer, Spring,
            Autumn, Winter
};
typedef enum Season SEASON;
```

OBJECTIVE

- Understanding how to define new variable with pre-defined values
- Creating new variable type and pre-defined values using **enum**

Defining variable with predefined values

❑ Introduction

Enumeration

Enumeration is a new data type and consisting a set of values defined by a programmer during creation.

Defining variable with predefined values

❑ Enumeration

- Enumeration is a new data type and consisting a set of values defined by a programmer during creation.
- The value can be name, month, day, color, or anything that could make a program code easy to read and maintain.
- **Example:**
  - Month of the year: `(January, February, March,..., December)`
  - Car's brand: `(Lexus, Mercedes Benz,...)`
  - Marital status: `(Single, Divorce, Married,...)`

Defining variable with predefined values

❑ Declaration

- To create an enumeration type:

```
enum identifierName {value1, value2, …, valueN}
```

Examples:

```
enum day {Monday, Tuesday,…, Sunday}

enum color {red, blue, white, black}
```

Defining variable with predefined values

❑ Variable of enumerated type

- Declaration:
  - `Var identifier : name of enumerated type`
  - `Var d: day`
- Usage:
  - Can only assign a constant to a variable of the enumeration or
  - Compare a variable of the type
- Example:

```
enum day {Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday}
Var d: day
d ← Monday
if (d == Wednesday) then
        …
end if
```

Defining variable with predefined values

❑ Conversion of enumerated type

- The value of enumerated type is considered as integer when display.
  - It is in the order start from 0
- Example:

```
var d: day
for(d ← Monday; d<=Sunday; d ← d + 1) do
     write(d)
end for
```

Defining variable with predefined values

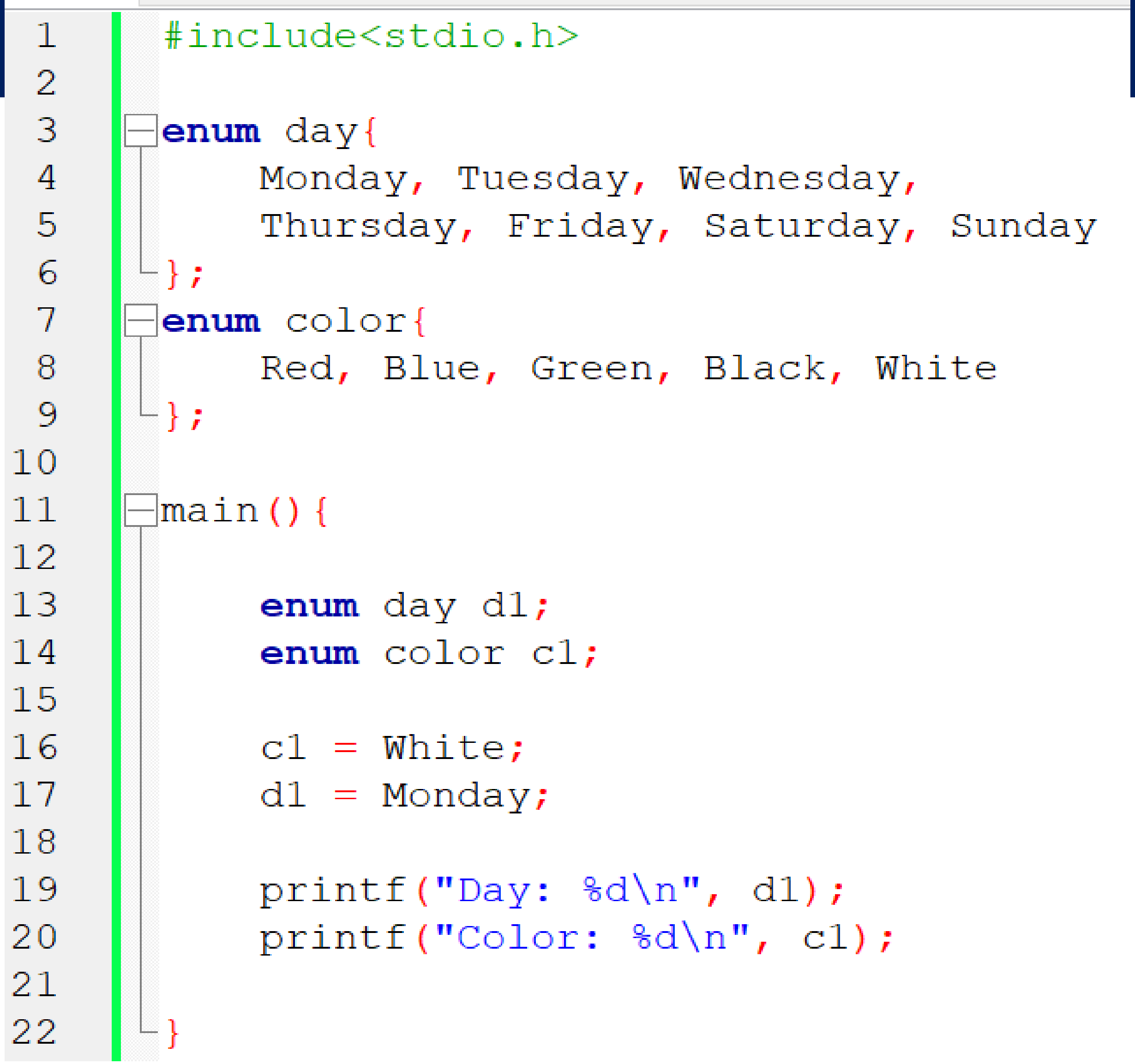
❑ Example: Using enumeration type

```
enum color{black, white, red, yellow, blue}
Procedure display_color(c: color)
       switch(c) do
              black: write(“Black color”)
              white: write(“white color”)
              red: write(“red color”)
              yellow: write(“blue color”)
       end switch
End procedure
```

```
Var tmp: color
begin
      tmp ← black
      display_color(tmp)
end
```

Defining variable with predefined values

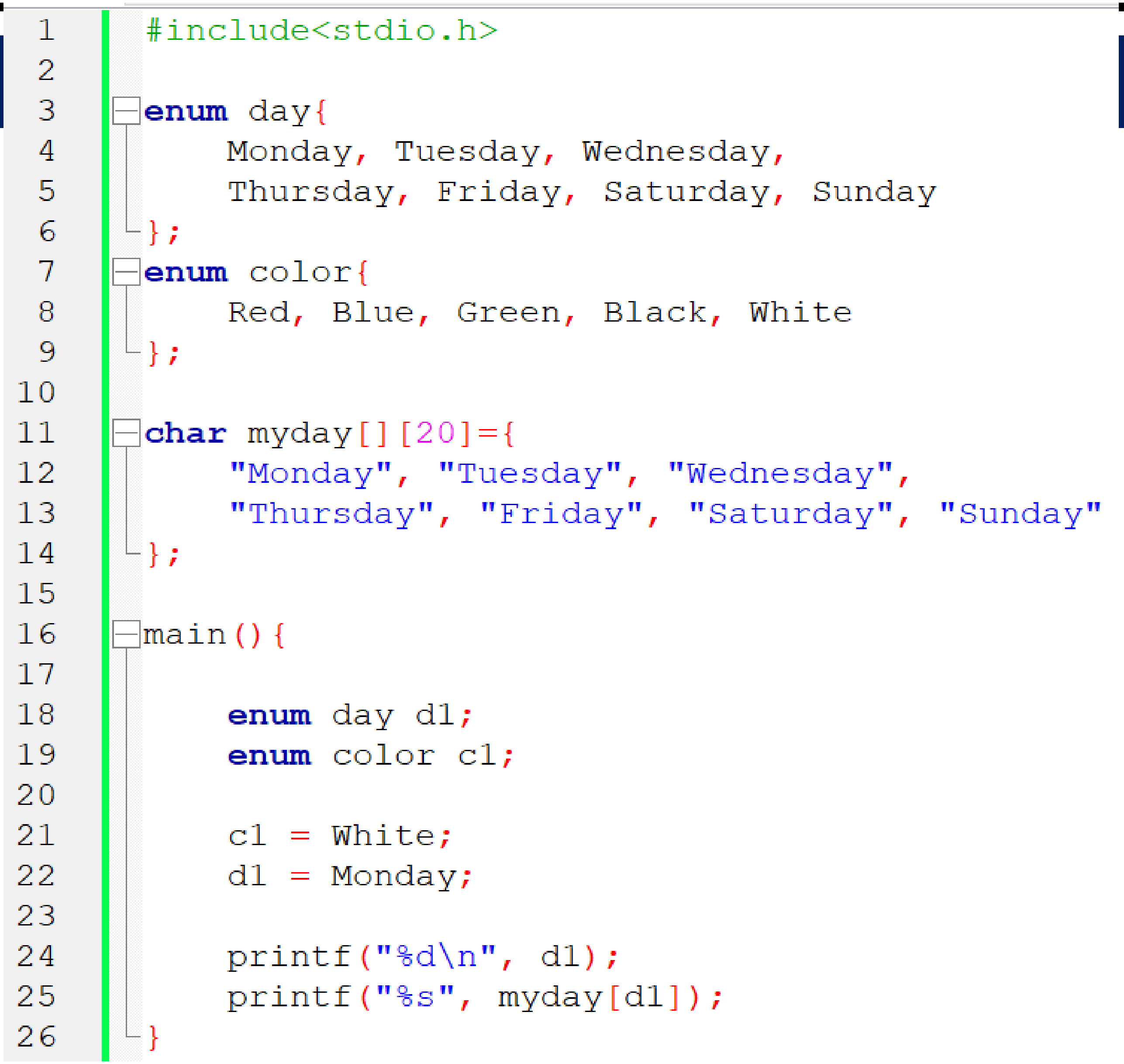
❑ Example 1



```
#include<stdio.h>

enum day{
    Monday, Tuesday, Wednesday,
    Thursday, Friday, Saturday, Sunday
};
enum color{
    Red, Blue, Green, Black, White
};

main(){

    enum day d1;
    enum color c1;

    c1 = White;
    d1 = Monday;

    printf("Day: %d\n", d1);
    printf("Color: %d\n", c1);

}
```

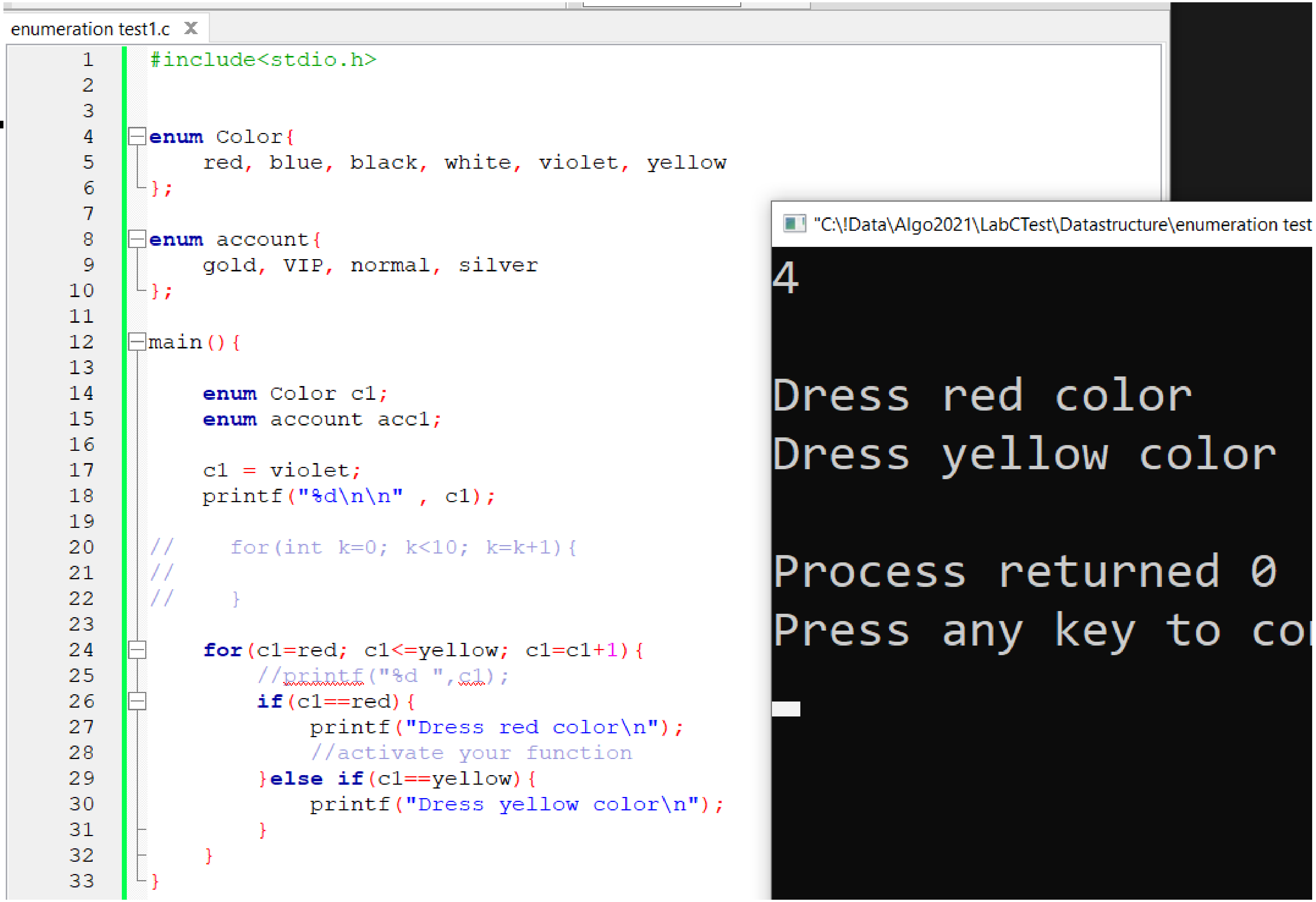
Defining variable with predefined values

- ❑ Example 2



```
#include<stdio.h>

enum day{
    Monday, Tuesday, Wednesday,
    Thursday, Friday, Saturday, Sunday
};
enum color{
    Red, Blue, Green, Black, White
};

char myday[][20]={
    "Monday", "Tuesday", "Wednesday",
    "Thursday", "Friday", "Saturday", "Sunday"
};

main(){

    enum day d1;
    enum color c1;

    c1 = White;
    d1 = Monday;

    printf("%d\n", d1);
    printf("%s", myday[d1]);
}
```

Examples

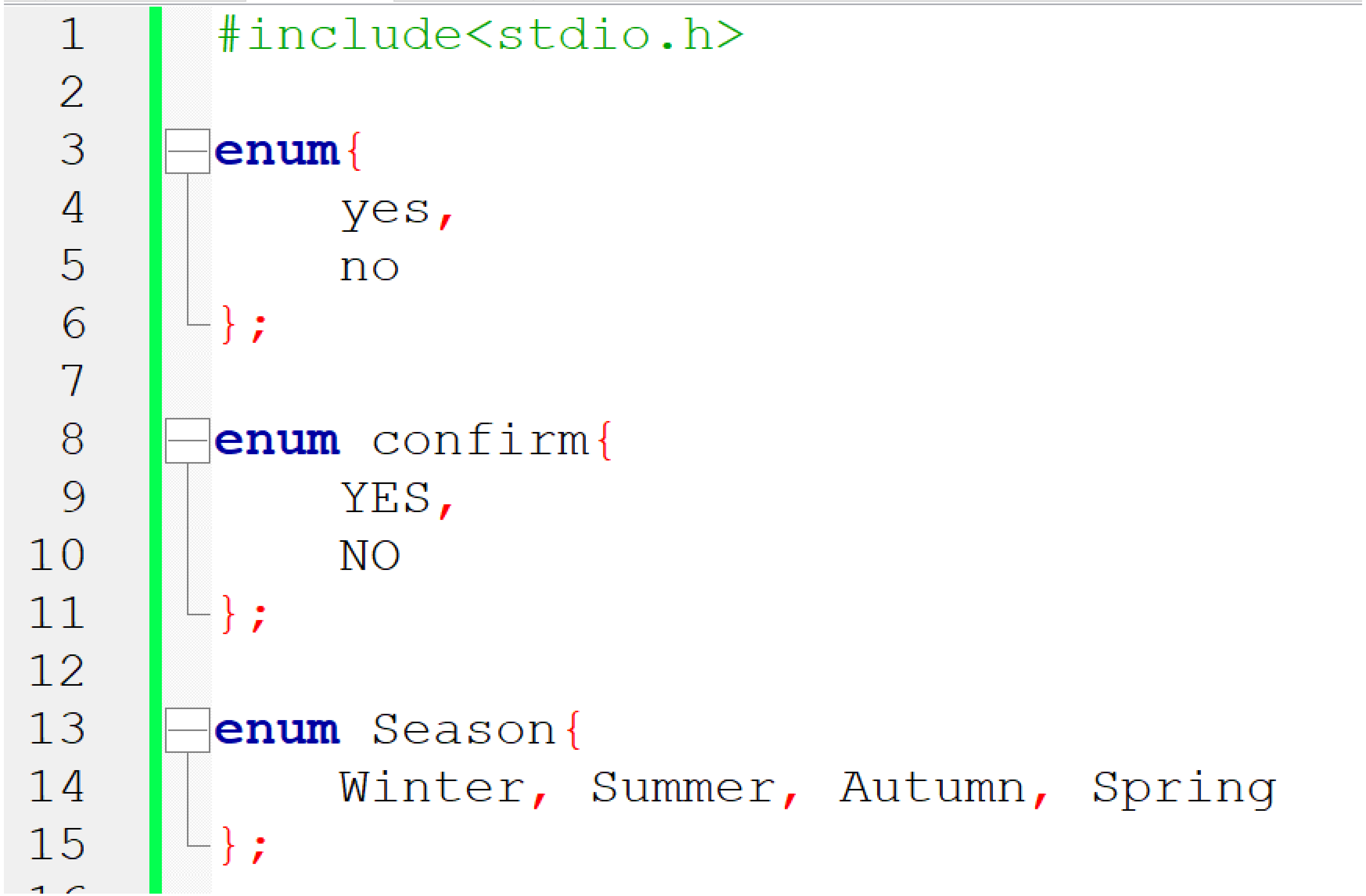


```c
#include<stdio.h>


enum Color{
    red, blue, black, white, violet, yellow
};

enum account{
    gold, VIP, normal, silver
};

main(){

    enum Color c1;
    enum account acc1;

    c1 = violet;
    printf("%d\n\n" , c1);

//    for(int k=0; k<10; k=k+1){
//
//    }

    for(c1=red; c1<=yellow; c1=c1+1){
        //printf("%d ",c1);
        if(c1==red){
            printf("Dress red color\n");
            //activate your function
        }else if(c1==yellow){
            printf("Dress yellow color\n");
        }
    }
}
```

"C:\!Data\Algo2021\LabCTest\Datastructure\enumeration test

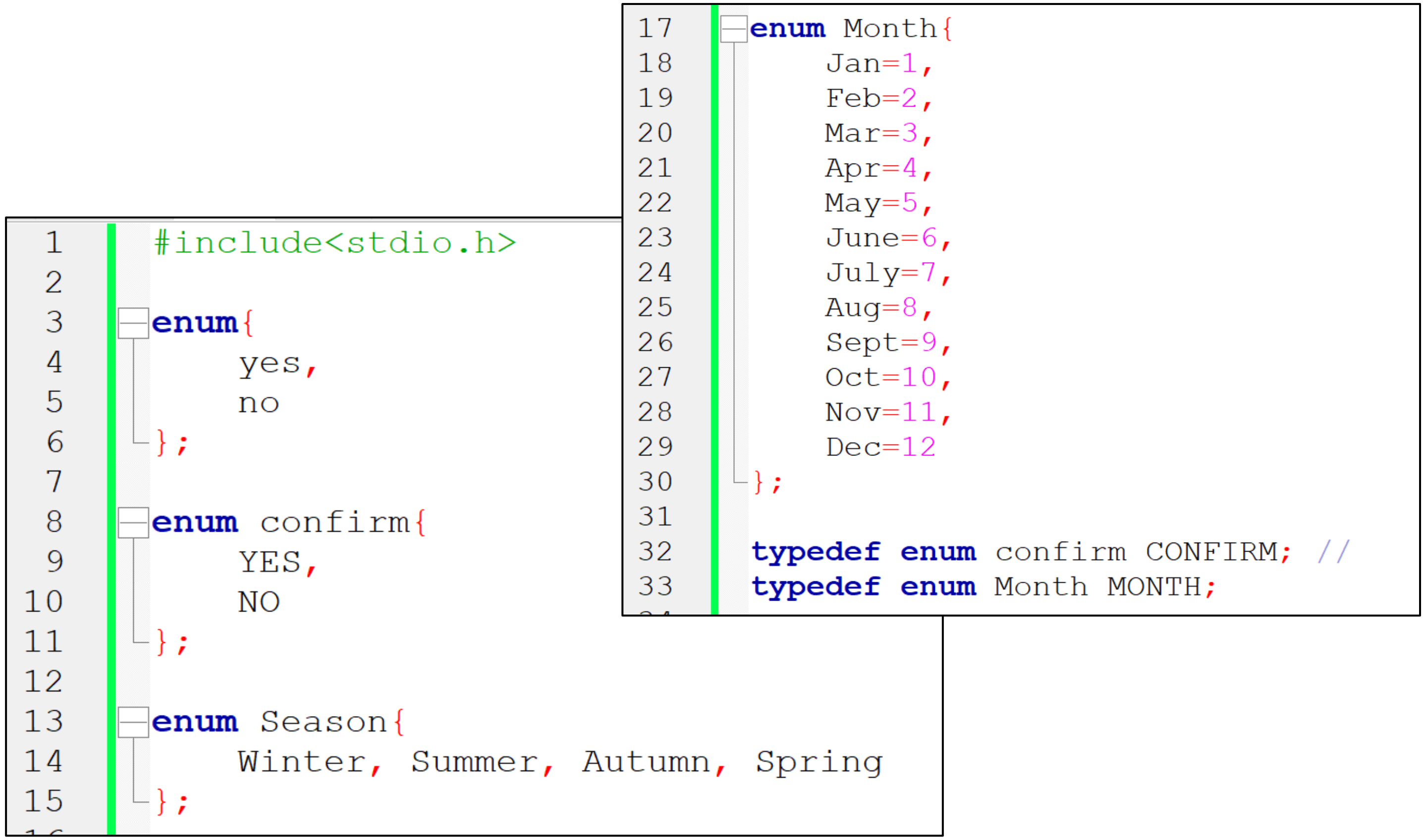
```
4

Dress red color
Dress yellow color

Process returned 0
Press any key to co
```

Examples



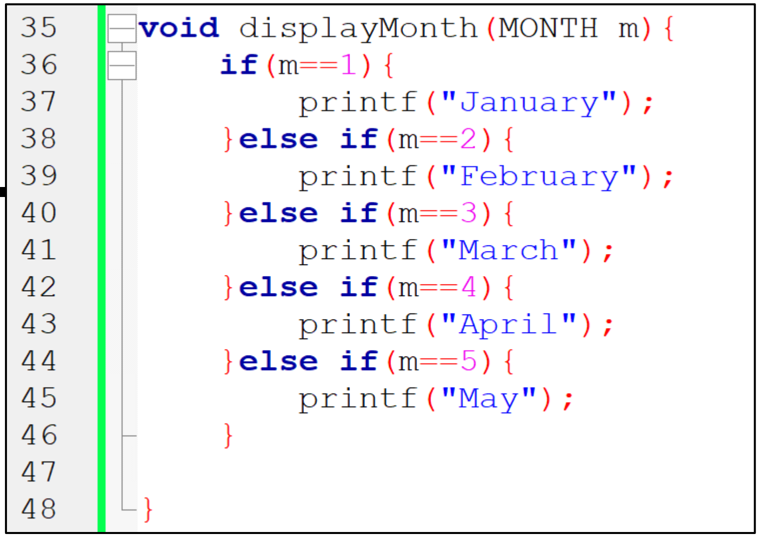
```c
#include<stdio.h>

enum{
    yes,
    no
};

enum confirm{
    YES,
    NO
};

enum Season{
    Winter, Summer, Autumn, Spring
};
```

Examples

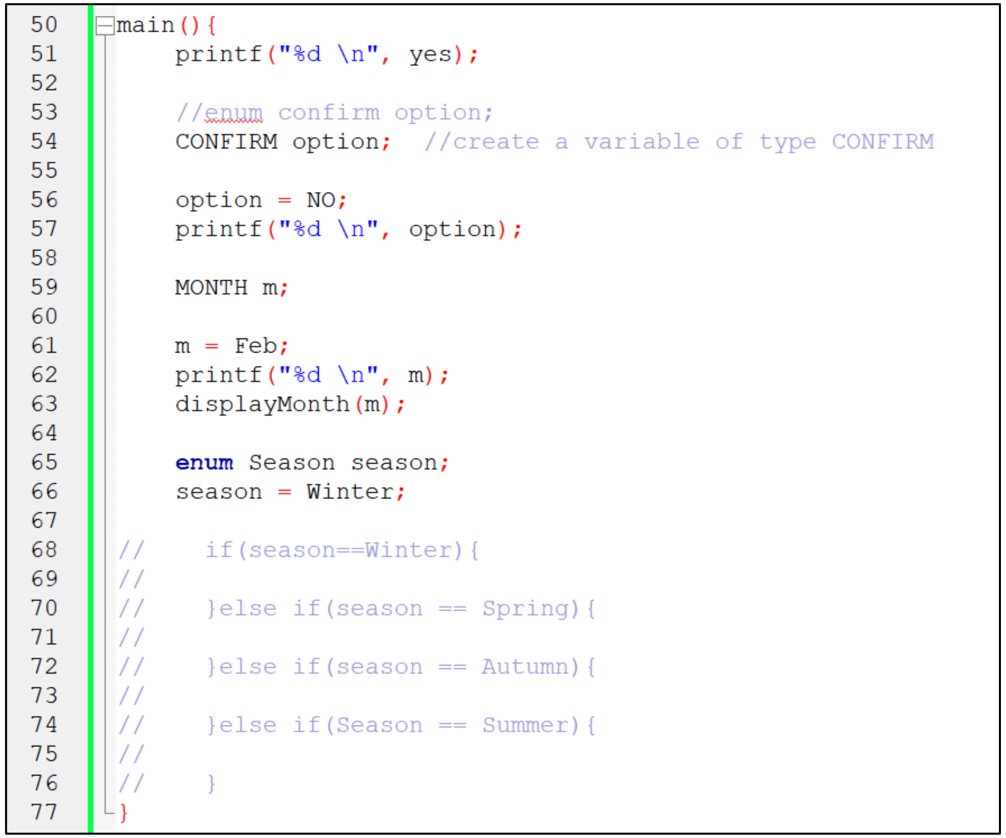


```c
#include<stdio.h>

enum{
    yes,
    no
};

enum confirm{
    YES,
    NO
};

enum Season{
    Winter, Summer, Autumn, Spring
};
```

```c
enum Month{
    Jan=1,
    Feb=2,
    Mar=3,
    Apr=4,
    May=5,
    June=6,
    July=7,
    Aug=8,
    Sept=9,
    Oct=10,
    Nov=11,
    Dec=12
};

typedef enum confirm CONFIRM; //
typedef enum Month MONTH;
```



```c
void displayMonth(MONTH m){
    if(m==1){
        printf("January");
    }else if(m==2){
        printf("February");
    }else if(m==3){
        printf("March");
    }else if(m==4){
        printf("April");
    }else if(m==5){
        printf("May");
    }

}
```



```c
main(){
    printf("%d \n", yes);

    //enum confirm option;
    CONFIRM option;  //create a variable of type CONFIRM

    option = NO;
    printf("%d \n", option);

    MONTH m;

    m = Feb;
    printf("%d \n", m);
    displayMonth(m);

    enum Season season;
    season = Winter;

//    if(season==Winter){
//
//    }else if(season == Spring){
//
//    }else if(season == Autumn){
//
//    }else if(Season == Summer){
//
//    }
}
```