\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Report: hw2

Author: F74055047 張財實 <f74055047@mail.ncku.edu.tw>

Class: 乙班

Description:

I have done this homework by referring the textbook chapter 6 project 8, after that a month-calendar is referred to do. Next, I figured out the pattern of a calendar, the key solution is to calculate the start day with both year and month.

This homework trains us to optimize a program using for if else switch. It helps a lot.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Code:

#include<stdio.h>

int main(int argc,char\*argv[]){

int year, month, day, thisyear, date, firstday, thismonth;

//firstday is start day

//thisyear is input year

//this month is input month

thisyear=atoi(argv[1]);

firstday = 1;//if firstday is in 1899

//then after the first loop,the firstday is of 1900

if(argc==2){

//only input year

if(thisyear<1900)//make sure thisyear is start at 1900

goto valid;//if it does not ,output warning

for(year=1899;year<thisyear;year++){

if((year % 400 == 0)||(year % 4 ==0 && year % 100 !=0))//formula for leap year

day=366; //leap year days of a year

else

day=365; //normal year days

firstday=(firstday+day)%7;//calender is made of 7 columns,so the remainder %7,end of the loop will outcome next year's firstday

}

for(month=1;month<=12;month++){

//loop of month specifically calculate first day of each month

switch(month){

//print each month and assignment each month's day magnitude

case 1: printf("January\n\n");day=31;break;

case 3: printf("March\n\n");day=31;break;

case 5: printf("May\n\n");day=31;break;

case 7: printf("July\n\n");day=31;break;

case 8: printf("August\n\n");day=31;break;

case 10: printf("October\n\n");day=31;break;

case 12: printf("December\n\n");day=31;break;

case 2:

printf("February\n\n");

//for leap year's February

if((year % 400 == 0)||(year % 4 ==0 && year % 100 !=0))

day=29;

else

day=28;

break;

case 4:printf("April\n\n");day = 30;break;

case 6:printf("June\n\n");day = 30;break;

case 9:printf("September\n\n");day = 30;break;

case 11:printf("November\n\n");day = 30;break;

}

printf("Sun Mon Tue Wed Thu Fri Sat\n");

//start counting days and refer to chp6 pj8 on ext book

//start of printing calender

if(firstday==0){

//in %7 ,if firstday==0 means firstday is on the 7th column of calender

//so that there is 6 blank days to fill

for(date=1;date<=6;date++){

if(date==1)

//first column is 3d

printf(" ");

else

printf(" ");}

for(date=1;date<=day;date++){

if((firstday+date-1)%7==1)

printf("%3d",date);

//first column is 3d

else

printf("%5d",date);

if((firstday+date-1)%7==0)

printf("\n");}

}

else{

//if firstday !=0,then just print the entire month normally

for(date=1;date<=(firstday-1);date++){

if(date==1)

printf(" ");

else

printf(" ");

}

for(date=1;date<=day;date++){

if((firstday+date-1)%7==1)

printf("%3d",date);

else

printf("%5d",date);

if((firstday+date-1)%7==0)

printf("\n");}

}

firstday=(firstday+day)%7;

printf("\n\n");

}

}

else if(argc==3){

//if user input year and month

thismonth=atoi(argv[2]);

if(thismonth>12||thismonth<1||thisyear<1900)

goto valid;

//either thisyear and thismonth is out of range the program will show warning to user

for(year=1899;year<thisyear;year++){

//loop of year to thisyear

if((year % 400 == 0)||(year % 4 ==0 && year % 100 !=0))

day=366;

else

day=365;

firstday=(firstday+day)%7;

}

for(month=1;month<=12;month++){

//then loop of month in thisyear

switch(month){

case 1:case 3:case 5:case 7:case 8:case 10:case 12:

day=31;break;

case 2:

if((year % 400 == 0)||(year % 4 ==0 && year % 100 !=0))

day=29;

else

day=28;

break;

case 4:case 6:case 9:case 11:

day=30;break;

}

if(month==thismonth){

//only if month matches with thismonth,printing starts from here

switch(month){

case 1:printf("January\n\n");break;

case 2:printf("February\n\n");break;

case 3:printf("March\n\n");break;

case 4:printf("April\n\n");break;

case 5:printf("May\n\n");break;

case 6:printf("June\n\n");break;

case 7:printf("July\n\n");break;

case 8:printf("August\n\n");break;

case 9:printf("September\n\n");break;

case 10:printf("October\n\n");break;

case 11:printf("November\n\n");break;

case 12:printf("December\n\n");break;

}

printf("Sun Mon Tue Wed Thu Fri Sat\n");

//same as above code

if(firstday==0){

for(date=1;date<=6;date++){

if(date==1)

printf(" ");

else

printf(" ");}

for(date=1;date<=day;date++){

if((firstday+date-1)%7==1)

printf("%3d",date);

else

printf("%5d",date);

if((firstday+date-1)%7==0)

printf("\n");}

}

else{

for(date=1;date<=(firstday-1);date++){

if(date==1)

printf(" ");

else

printf(" ");

}

for(date=1;date<=day;date++){

if((firstday+date-1)%7==1)

printf("%3d",date);

else

printf("%5d",date);

if((firstday+date-1)%7==0)

printf("\n");}

}

}

firstday=(firstday+day)%7;

}

printf("\n");

}

else

valid: printf("Please enter valid input\n");//the warning

}

Compilation:

gcc -o hw2 hw2.c

Execution:

1)./hw2 2010

2)./hw2 2010 10

Output:

1)

January

Sun Mon Tue Wed Thu Fri Sat

1 2

3 4 5 6 7 8 9

10 11 12 13 14 15 16

17 18 19 20 21 22 23

24 25 26 27 28 29 30

31

February

Sun Mon Tue Wed Thu Fri Sat

1 2 3 4 5 6

7 8 9 10 11 12 13

14 15 16 17 18 19 20

21 22 23 24 25 26 27

28

March

Sun Mon Tue Wed Thu Fri Sat

1 2 3 4 5 6

7 8 9 10 11 12 13

14 15 16 17 18 19 20

21 22 23 24 25 26 27

28 29 30 31

April

Sun Mon Tue Wed Thu Fri Sat

1 2 3

4 5 6 7 8 9 10

11 12 13 14 15 16 17

18 19 20 21 22 23 24

25 26 27 28 29 30

May

Sun Mon Tue Wed Thu Fri Sat

1

2 3 4 5 6 7 8

9 10 11 12 13 14 15

16 17 18 19 20 21 22

23 24 25 26 27 28 29

30 31

June

Sun Mon Tue Wed Thu Fri Sat

1 2 3 4 5

6 7 8 9 10 11 12

13 14 15 16 17 18 19

20 21 22 23 24 25 26

27 28 29 30

July

Sun Mon Tue Wed Thu Fri Sat

1 2 3

4 5 6 7 8 9 10

11 12 13 14 15 16 17

18 19 20 21 22 23 24

25 26 27 28 29 30 31

August

Sun Mon Tue Wed Thu Fri Sat

1 2 3 4 5 6 7

8 9 10 11 12 13 14

15 16 17 18 19 20 21

22 23 24 25 26 27 28

29 30 31

September

Sun Mon Tue Wed Thu Fri Sat

1 2 3 4

5 6 7 8 9 10 11

12 13 14 15 16 17 18

19 20 21 22 23 24 25

26 27 28 29 30

October

Sun Mon Tue Wed Thu Fri Sat

1 2

3 4 5 6 7 8 9

10 11 12 13 14 15 16

17 18 19 20 21 22 23

24 25 26 27 28 29 30

31

November

Sun Mon Tue Wed Thu Fri Sat

1 2 3 4 5 6

7 8 9 10 11 12 13

14 15 16 17 18 19 20

21 22 23 24 25 26 27

28 29 30

December

Sun Mon Tue Wed Thu Fri Sat

1 2 3 4

5 6 7 8 9 10 11

12 13 14 15 16 17 18

19 20 21 22 23 24 25

26 27 28 29 30 31

2)  
October

Sun Mon Tue Wed Thu Fri Sat

1 2

3 4 5 6 7 8 9

10 11 12 13 14 15 16

17 18 19 20 21 22 23

24 25 26 27 28 29 30

31