

# LAB 2 - TASK 1 through TASK 5

## vi / Constructs / Age / Stacks / Millionaire

John Dempsey

COMP-232: Programming Languages  
California State University, Channel Islands

August 30, 2023

Hard Due Date: September 6, 2023

Soft Due Date: September 20, 2023

For our second lab, we will write several short C programs. During Lab 3, I'll explain how to turn in your programs.

### TASK 1 – A Visual Editor (vi) Called vim

Type the vi commands that are in **bold font**.

1. Open an Ubuntu command window.
2. At the prompt, type: **vi welcome.txt**
3. Type **i** to enter INSERT MODE.
4. Type in the following sentence:

**Welcome to COMP-232 Programming Languages**

5. Hit **ESC** key to exit INSERT MODE.
6. Type **y** to yank the sentence.
7. Type **p** nine times to paste the above line in your welcome.txt file nine times.
8. Go to the top of the file by typing **1G** (or **gg**)
9. Hit **ENTER** key 5 times or type **6G** to go to the 6<sup>th</sup> line.
10. Type **i** again to enter INSERT MODE.
11. Type:

**vi is easy once you learn it.**

12. Hit **ESC** to exit INSERT MODE.
13. Type **G** to go to the bottom of the welcome.txt file.
14. Go to the end of the last line by typing **\$** sign
15. Type **a** to append the following to the end of the line:

**--- This is the last line.**

16. Hit **ESC** to exit INSERT MODE.
17. Type **/easy** to search for the word easy.
18. Change one word "easy" to "very easy" by typing:  
**cw** (for change word)

**very easy**

Hit **ESC** to exit INSERT MODE.

19. To save your file and exit vi, type **ZZ**.

You could also type **:wq** too: w to write file contents out, q to quit.

20. Edit welcome.txt again by typing:

**vi welcome.txt**

21. Find “very easy” by typing: **/very easy**

22. Now type **a** for append and type the word **one**. Hit **ESC** to exit INSERT MODE.

23. Now type **a** again for append and type the word **two**. Hit **ESC** to exit INSERT MODE.

24. Now press **u** to undo **two**.

25. Press **u** again to undo **one**.

26. Exit vi without saving any changes by typing: **:q!**

## TASK 2 Basic C Constructs

### Task 2.1 myfor.c - Use a for loop to print "Hello World!" 5 times.

Using vi, edit a file called myfor.c by typing:

```
% vi myfor.c
```

Edit file for.c to contain:

```
int main()
{
    for (i=0; i<5; i++)
        printf("Hello World!\n");
}
```

To compile myfor.c, run:

```
% gcc myfor.c -o myfor
```

To run myfor executable, run:

```
% myfor
```

### TASK 2.2 mywhile.c - Use a while loop to print "Hello World!" 10 times.

Copy the myfor.c to mywhile.c using:

```
% cp myfor.c mywhile.c
```

Using vi, edit mywhile.c to print Hello World ten times using a while loop instead of a for loop.

### TASK 2.3 myswitch.c - Switch/Case

Edit a file named myswitch.c.

Inside a while loop that never exits, read a character using getch().

If the character entered is:

1 – print "One for the money."

2 – print "Two for the show."

3 – print "Three to get ready"

4 – print "Four for the show"

q or Q – print "Thanks for playing." You can use break statement to exit while loop

Anything else – print “Invalid input. Please try again.”

Here’s a rough start:

```
#include <stdio.h>

int main()
{
    char    ch;

    while (1) {

        ch = getchar();

        switch (ch) {

            case '1': printf("One for the money.\n");
            case '2': printf("Two for the show\n");
            ....

        }

    }
}
```

Copy/paste the above code into your myswitch.c file and then type:

```
gg=G
```

to auto format your code.

You will also need to change the curly quotes Word uses to regular quotes too. Type:

```
:1,$s/"/"/g
```

```
:1,$s/'/'/g
```

```
:1,$s/'/'/g
```

The **:1,\$** means “for lines 1 through 10”.

**s/"/"/** means switch the curly quotes to regular quotes.

And the **g** stands for global, meaning change all curly quotes in the file.

## Task 2.4 myif.c – Use one large if / else statement

Copy myswitch.c to myif.c and replace the switch/case statement to use one large if / else statement instead.

The output should be the same as for Task 2.3.

### TASK 3 – How Old Are You?

This task calculates how old you are in years, months, days, hours, minutes, and seconds.

We'll assume no leap years and your age will be read in as a real number, like 19.40 years old. The .40 would represent #days/365 since your last birthday.

Here's a template which you can start with if you like:

```
john@oho:~/LAB2$ cat age.c
#include <stdio.h>

void main()
{
    float age;

    printf("How old are you? ");
    scanf("%f", &age);

    printf("You are %.02f years old.\n", age);
}
```

Now update the above program to calculate your age in years, months, days, hours, minutes, and seconds.

```
john@oho:~/LAB2$ cat age.c

john@oho:~/LAB2$ gcc age.c -o age

john@oho:~/LAB2$ age
How old are you? 19.6
You are 19.6 years old.
You are xxx months old.
You are xxx days old.
You are xxx hours old.
You are xxx minutes old.
You are xxx seconds old.
```

## TASK 4. Stacks

The stack.c program below almost works, but I must have been up too late and made a bunch of silly errors.

Copy and paste the stack.c program into stack.c on your computer and fix the errors. After pasting the stack.c program onto your Ubuntu system, format your program by running “**gg=G**” in vi.

Pay attention to the basics on how the stack works. Programming languages like to push items onto the stack, then pop a few items, modify the popped data, and then push the result back onto the stack.

```
john@oho:~/LAB2$ cat stack.c
#include <stdio.h>
#include <string.h>

#define MAX_STACK_SIZE 8
#define MAX_LENGTH 50

char data
char stack[MAX_STACK_SIZE][MAX_LENGTH];
int top = -1;

int isempty() {
    if(top == -1)
        return 1;
    else
        return 0;
}

int isfull() {
    if(top == MAX_STACK_SIZE)
        return 1;
    else
        return 0;
}

char *peek() {
    return stack[top];
}

char *pop() {
    if(!isempty()) {
        strcpy(data, stack[top]);
        top = top - 1;
        if (top == -1)
            printf("Pop \"%d\" off of stack. Stack is empty.\n", data);
        else
```

```

        printf("Pop \"%s\" off of stack. Top of stack contains: %s\n", data, stack[top]);
        return data;
    } else {
        printf("Error: Could not pop data. Stack is empty.\n");
    }
}

char *push(char *data) {
    if(!isfull()) {
        top = top + 1;
        data = stack[top];
        printf("Push \"%s\" onto stack.\n", data);
    } else {
        printf("Error: Could not push \"%s\" onto stack. Stack is full.\n", data);
    }
}

int main() {
    push("Islands");
    push("Channel");
    printf("There are %d items on stack. Top of stack is: %s\n", top+1, peek());

    push("San");
    push("Diego");
    printf("There are %d items on stack. Top of stack is: %s\n", top+1, peek());

    pop();
    pop();
    printf("There are %d items on stack. Top of stack is: %s\n", top+1, peek());

    push("to CSU");
    push("Welcome");
    printf("There are %s items on stack. Top of stack is: %d\n", top+1, peek());

    printf("All done adding.\n");

    printf("Value at the top of the stack is: %s\n", peek());

    printf("Pop everything off stack.\n");
    printf("-----\n");

    while(!isempty()) {
        pop();
        printf("%s\n", data);
    }
    printf("-----\n");
    return 0;
}

```

## TASK 5. Who Wants to Be a Millionaire?

As you get closer to your graduation from Channel Islands, the last thing on your mind is, well, retirement. But you've heard about the power of **compound interest** and you want to learn more. You want to know if contributing to your 401k can make you a millionaire?

To run the retire program, there are two input parameters:

1. How old you are when you start to save?
2. How much you consistently save per month until your retirement age of 65?

Copy/Paste the following retire.c template to start with:

```
john@oho:~/232/LAB2$ cat retire.c
#include <stdio.h>
#include <stdlib.h>

#define SALARY 100000 // $100K/year

void main(int argc, char *argv[])
{
    int after_tax_distribution = 0;
    int age = 0;
    int amount = 0;
    int distribution = 0;
    int i = 0;
    int matching_401k_amount = 0;
    int max_matching_401k_amount = 0;
    int principle = 0;
    int roth_distribution = 0; // Need to determine.
    int taxes = 0;
    int taxable_distribution;
    int total_saved = 0;
    int yearly_401k_contribution = 0;

    if (argc < 2) {
        printf("Missing input parameters to retire program.\n");
        printf("Usage: retire age_starting dollar_saved_per_month\n");
        exit(0);
    }

    age = atoi(argv[1]);
    amount =

    printf("\n");
    printf("Welcome to Your Retirement Calculator\n");
    printf("-----\n");
    printf(" 1. You start saving at age %d.\n", age);
    printf(" 2. The amount saved per month is $%d.\n", amount);
    printf("-----\n");

    for (i=age;i<65;i++) {
        yearly_401k_contribution = 12 * amount;
```



```

principle = principle + yearly_401k_contribution;

// 6% of salary is matched at 50%.
max_matching_401k_amount = SALARY * .06;

if (yearly_401k_contribution >= max_matching_401k_amount)
    matching_401k_amount = 0;          // NEED TO CORRECT
else
    matching_401k_amount = 0;          // NEED TO CORRECT

// 1.107 is 10.7%, the average gain over 30 years in an S&P Index Fund.
total_saved = (total_saved + (12 * amount)) * 1.107 + matching_401k_amount;

printf("At age %d, principle = %d, total saved = %d\n",
        i, principle, total_saved);
}

printf("-----\n");
printf("From age %d to 65, if %d is saved per month:\n", age, amount);
printf("    The principle amount you contributed is : %d\n", principle);
printf("    The principle amount + your gains is      : %d\n", total_saved);
printf("-----\n");

printf("\n");
printf("-----\n");
printf("But how much money will I receive per year from 65 to 95?\n");
printf("-----\n");

before_tax_distribution = total_saved/(95-65);

// THE IF STATEMENTS BELOW ARE BASED ON 2022 TAX TABLES.
// NEED TO UPDATE AND CORRECT FOR 2023 TAX TABLES.

if (distribution <= 10275)
    taxes = distribution * 0.10;          // 10%
else if (distribution <= 41775)
    taxes = 0;                          // 12%
else if (distribution <= 89075)
    taxes = (10275*0.10) + (41775-10275)*.12 + (distribution-41775)*.22; // 22%
else if (distribution <= 170050)
    taxes = 0;                          // 24%
else if (distribution <= 215950)
    taxes = 0;                          // 32%
else if (distribution <= 539900)
    taxes = 0;                          // 35%
else
    taxes = 0;                          // 37%

after_tax_distribution = before_tax_distribution - taxes;
roth_distribution = ;

printf("Distribution before taxes is: %d\n",      before_tax_distribution);
printf("Distribution from a 401k account is: %d\n", after_tax_distribution);
printf("Distribution from a ROTH account is: %d\n", roth_distribution);
printf("-----\n");
}

```

To make life easier, we'll define all variables as integers.

## Marginal tax brackets for tax year 2023, single individuals

Taxable income	Taxes owed
\$11,000 or less	10% of the taxable income
\$11,001 to \$44,725	\$1,100 plus 12% of amount over \$11,000
\$44,726 to \$95,375	\$5,147 plus 22% of amount over \$44,725
\$95,376 to \$182,100	\$16,290 plus 24% of amount over \$95,375
\$182,101 to \$231,250	\$37,104 plus 32% of amount over \$182,100
\$231,251 to \$578,125	\$52,832 plus 35% of amount over \$231,250
\$578,126 or more	\$174,238.25 plus 37% of amount over \$578,125

Below are our assumptions. This is not financial advice.

1. You retire on your birthday at the age of 65.
2. You invest in an S&P 500 Index Fund which we'll assume earns 10.7% per year.
3. You base the 10.7% yearly gain knowing that over the past 30 years, the S&P 500 index has delivered a compound average annual growth rate of 10.7% per year.
4. Your employer matches 50% of your 401k contribution up to 6% of your salary.
5. Your employer does "End of Year" 401k Matching. (Not great for you as the employee.)
6. Your lifetime salary is \$100,000/year and you never get a raise!
7. To reduce losing your money in retirement, you move the full amount saved on your birthday at the age of 65 into a money market account that earns 0% interest.
8. You receive the same distribution amount each year from 65 to 95 based on the total saved while working.

9. You stay single.

10. Since the amount in your 401k account is taxable, we will use the 2023 Tax Brackets for 30 years. We will ignore California state taxes that tax 9% of your salary.

Below is an example of running the retire program for the wrong tax tables. 60 represents the person's age when they start to save money. The value of 100 is the amount saved per month (i.e., \$100/mo).

**% retire 60 100**

Welcome to Your Retirement Calculator

- 
1. You start saving at age 60.
  2. The amount saved per month is \$100.
- 

At age 60, principle = \$1200, total saved = \$1928  
At age 61, principle = \$2400, total saved = \$4062  
At age 62, principle = \$3600, total saved = \$6425  
At age 63, principle = \$4800, total saved = \$9040  
At age 64, principle = \$6000, total saved = \$11935

---

From age 60 to 65, if \$100 is saved per month:

The principle amount saved is : \$6000  
The principle + gains amount is: \$11935

---

---

But how much money will I receive per year from 65 to 95?

---

Distribution before taxes is: \$397  
Distribution if a 401k account is : \$358  
Distribution if a ROTH account is : \$ Need to figure out.

---

**To get full credit for Task 4, please run the retire program using the following commands.**

**% script 401k.txt**

**% retire 25 100**

**% retire 35 100**

**% retire 55 1000**

**% quit**

**Answer the following two questions:**

**Question 1.**

**If you delay your contributions into your 401k by 10 years (i.e., you start at age 35 instead of 25), how much less money do you save?**

**Question 2:**

**Starting at age 25, how much money do you need to contribute to reach \$2.5 million dollars?**

**During LAB 3, I'll explain how to turn in your LAB 2 work during the lab.**