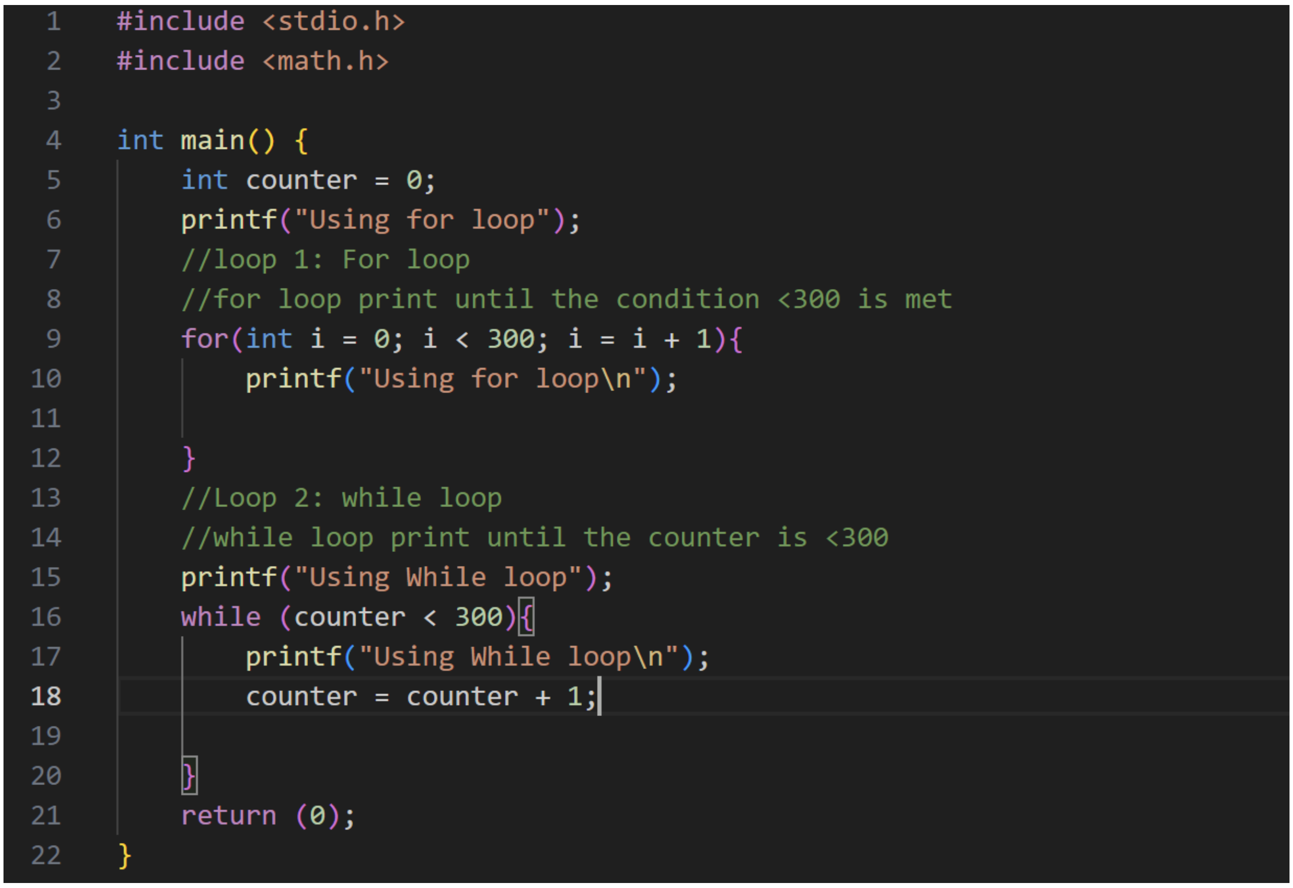
FIT3143 2024 - Applied Session Week 2

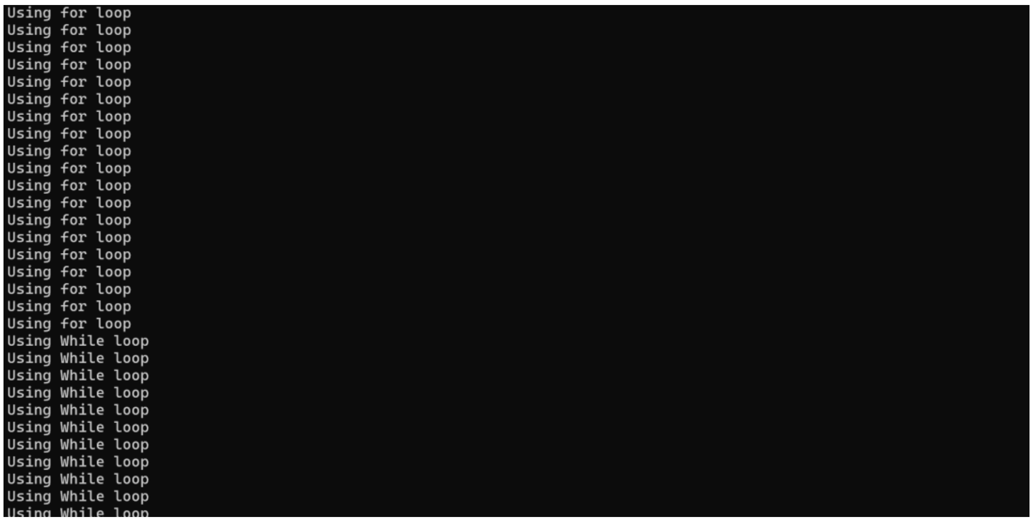
Student ID	**Student Name**	**Student Email**
32844700	Teh Jia Xuan	Jteh0015@student.monash .edu

Task 3



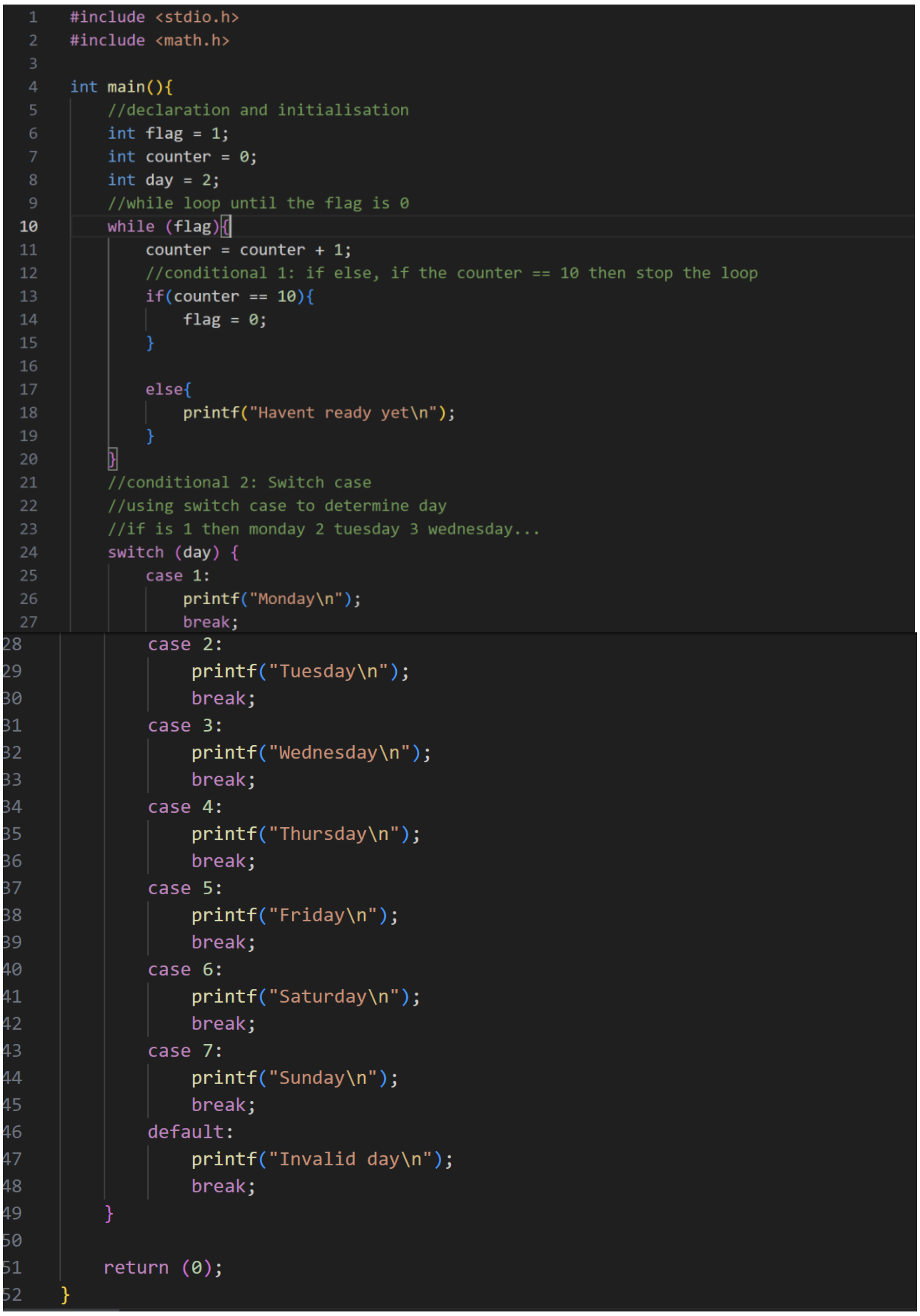
```c
#include <stdio.h>
#include <math.h>

int main() {
    int counter = 0;
    printf("Using for loop");
    //loop 1: For loop
    //for loop print until the condition <300 is met
    for(int i = 0; i < 300; i = i + 1){
        printf("Using for loop\n");

    }
    //Loop 2: while loop
    //while loop print until the counter is <300
    printf("Using While loop");
    while (counter < 300){
        printf("Using While loop\n");
        counter = counter + 1;

    }
    return (0);
}
```

Result:



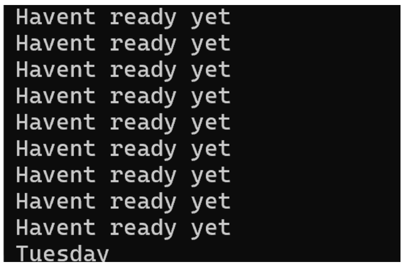
```
Using for loop
Using for loop
Using for loop
Using for loop
Using for loop
Using for loop
Using for loop
Using for loop
Using for loop
Using for loop
Using for loop
Using for loop
Using for loop
Using for loop
Using for loop
Using for loop
Using for loop
Using for loop
Using for loop
Using While loop
Using While loop
Using While loop
Using While loop
Using While loop
Using While loop
Using While loop
Using While loop
Using While loop
Using While loop
Using While loop
```

Task 5



```c
#include <stdio.h>
#include <math.h>

int main(){
    //declaration and initialisation
    int flag = 1;
    int counter = 0;
    int day = 2;
    //while loop until the flag is 0
    while (flag){
        counter = counter + 1;
        //conditional 1: if else, if the counter == 10 then stop the loop
        if(counter == 10){
            flag = 0;
        }

        else{
            printf("Havent ready yet\n");
        }
    }
    //conditional 2: Switch case
    //using switch case to determine day
    //if is 1 then monday 2 tuesday 3 wednesday...
    switch (day) {
        case 1:
            printf("Monday\n");
            break;
        case 2:
            printf("Tuesday\n");
            break;
        case 3:
            printf("Wednesday\n");
            break;
        case 4:
            printf("Thursday\n");
            break;
        case 5:
            printf("Friday\n");
            break;
        case 6:
            printf("Saturday\n");
            break;
        case 7:
            printf("Sunday\n");
            break;
        default:
            printf("Invalid day\n");
            break;
    }

    return (0);
}
```

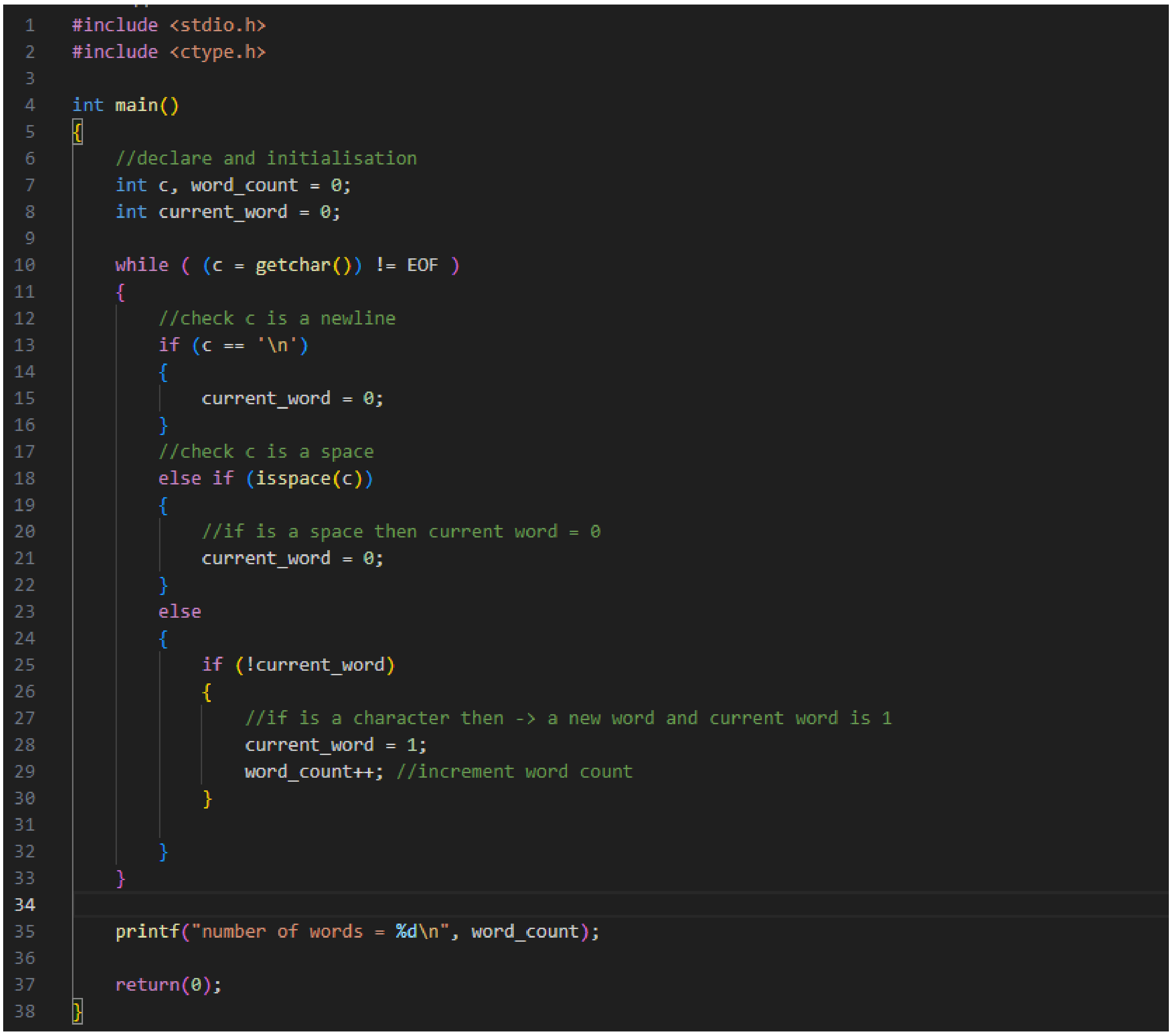
Result:



```
Havent ready yet
Havent ready yet
Havent ready yet
Havent ready yet
Havent ready yet
Havent ready yet
Havent ready yet
Havent ready yet
Havent ready yet
Tuesday
```

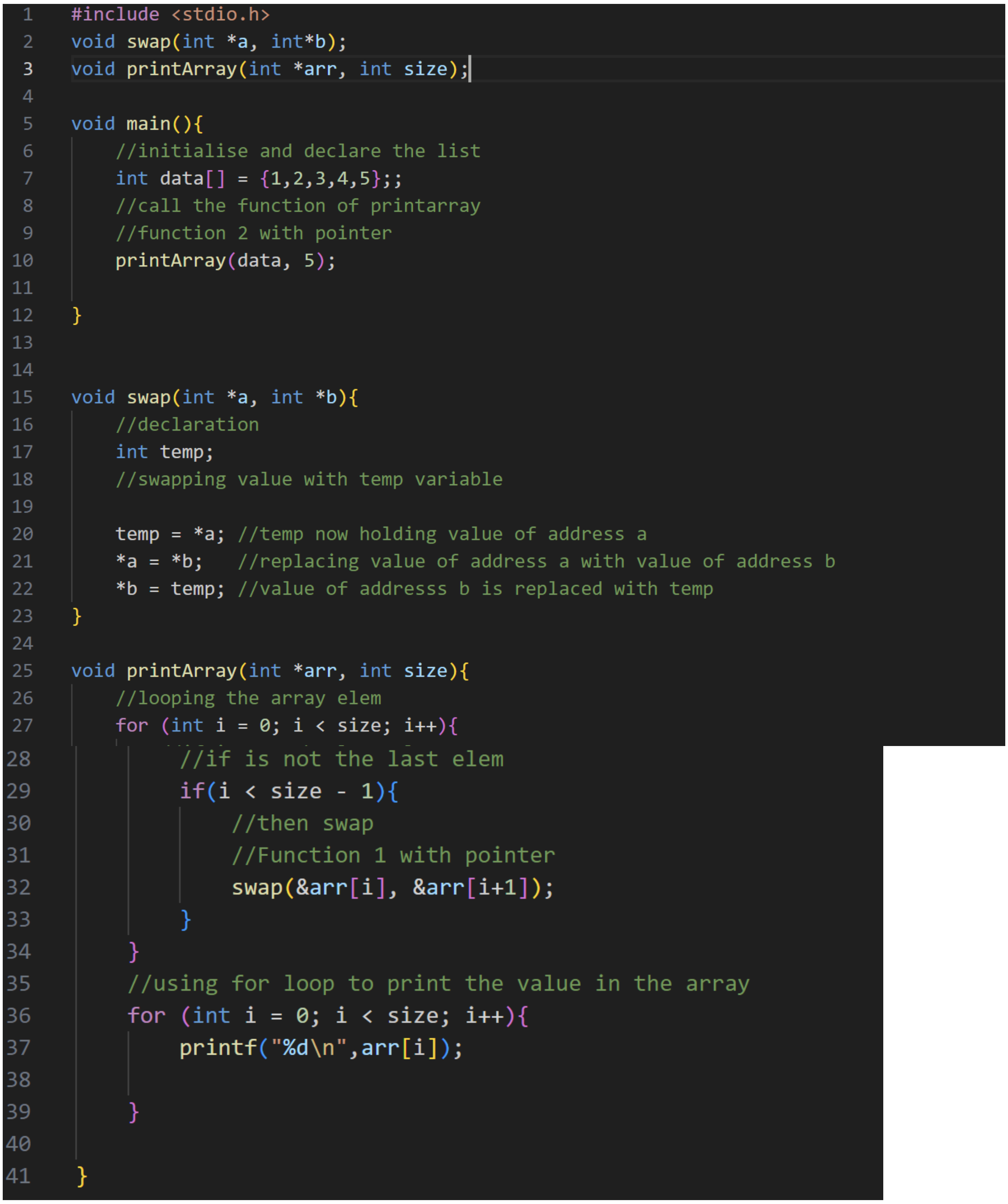
Task 7

We can use while ((c = getchar()) != EOF) to continuously reads characters until the end of file EOF. We can initialise a counter to count the words, we also declare a variable called current_word to count the words that is not using space to separate. For example hi\nhi (EOF). So whenever we encounter space or a new line we set current_word to 0. Then when c is a character and current_word is 0 then we set current_word to 1 and increment the word counter. So every time we need the current_word to be 0 so we can increment the word counter. Thus, every time we encounter space or new line we set current_word to 0.



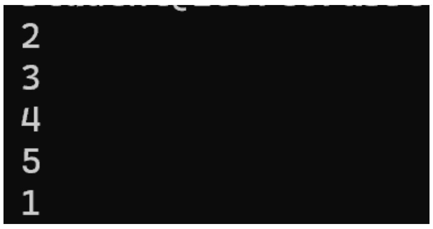
```c
#include <stdio.h>
#include <ctype.h>

int main()
{
    //declare and initialisation
    int c, word_count = 0;
    int current_word = 0;

    while ( (c = getchar()) != EOF )
    {
        //check c is a newline
        if (c == '\n')
        {
            current_word = 0;
        }
        //check c is a space
        else if (isspace(c))
        {
            //if is a space then current word = 0
            current_word = 0;
        }
        else
        {
            if (!current_word)
            {
                //if is a character then -> a new word and current word is 1
                current_word = 1;
                word_count++; //increment word count
            }

        }
    }

    printf("number of words = %d\n", word_count);

    return(0);
}
```

Task 11



```
#include <stdio.h>
void swap(int *a, int*b);
void printArray(int *arr, int size);

void main(){
    //initialise and declare the list
    int data[] = {1,2,3,4,5};;
    //call the function of printarray
    //function 2 with pointer
    printArray(data, 5);

}


void swap(int *a, int *b){
    //declaration
    int temp;
    //swapping value with temp variable

    temp = *a; //temp now holding value of address a
    *a = *b;   //replacing value of address a with value of address b
    *b = temp; //value of addresss b is replaced with temp
}

void printArray(int *arr, int size){
    //looping the array elem
    for (int i = 0; i < size; i++){
        //if is not the last elem
        if(i < size - 1){
            //then swap
            //Function 1 with pointer
            swap(&arr[i], &arr[i+1]);
        }
    }
    //using for loop to print the value in the array
    for (int i = 0; i < size; i++){
        printf("%d\n",arr[i]);

    }

}
```

Result:



```
2
3
4
5
1
```

Task 13

Initially BValue is declared as a pointer by this line “int AValue, *BValue;”. Then AValue is initialised with the value 101. Then BValue pointer is set to AValue’s address. So now BValue is pointing to AValue’s address. After that, BValue is dereferencing using *BValue, so it accessed the value stored at the address the pointer is pointing to, which is AValue’s address so BValue returns 101.