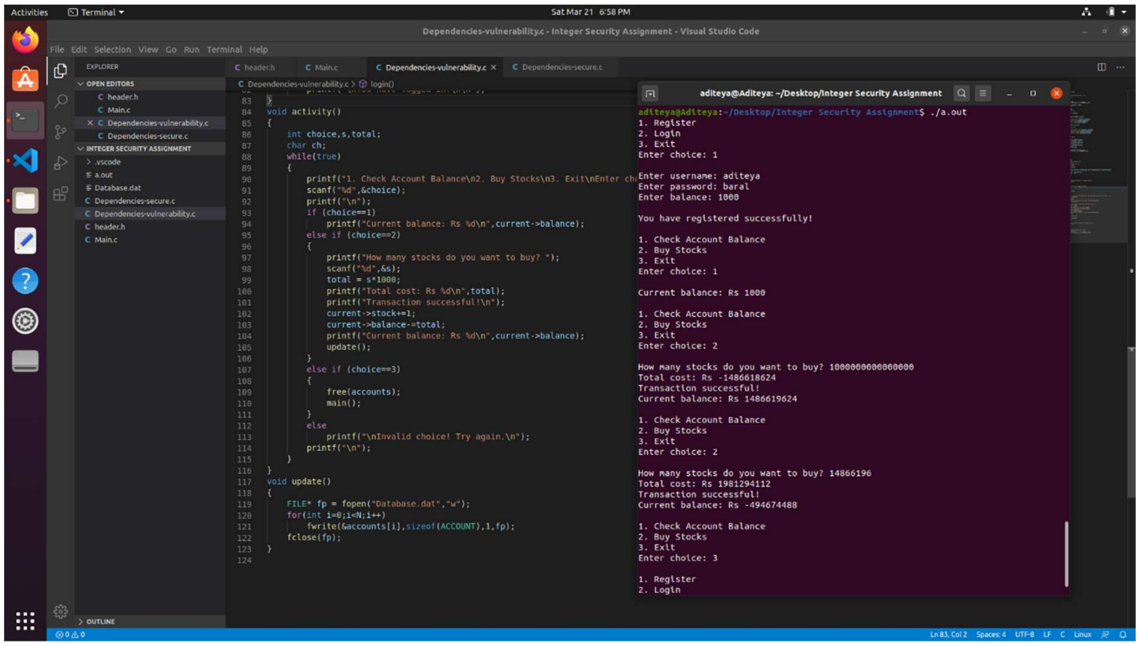
Integer Security Demo

Vulnerability Condition:

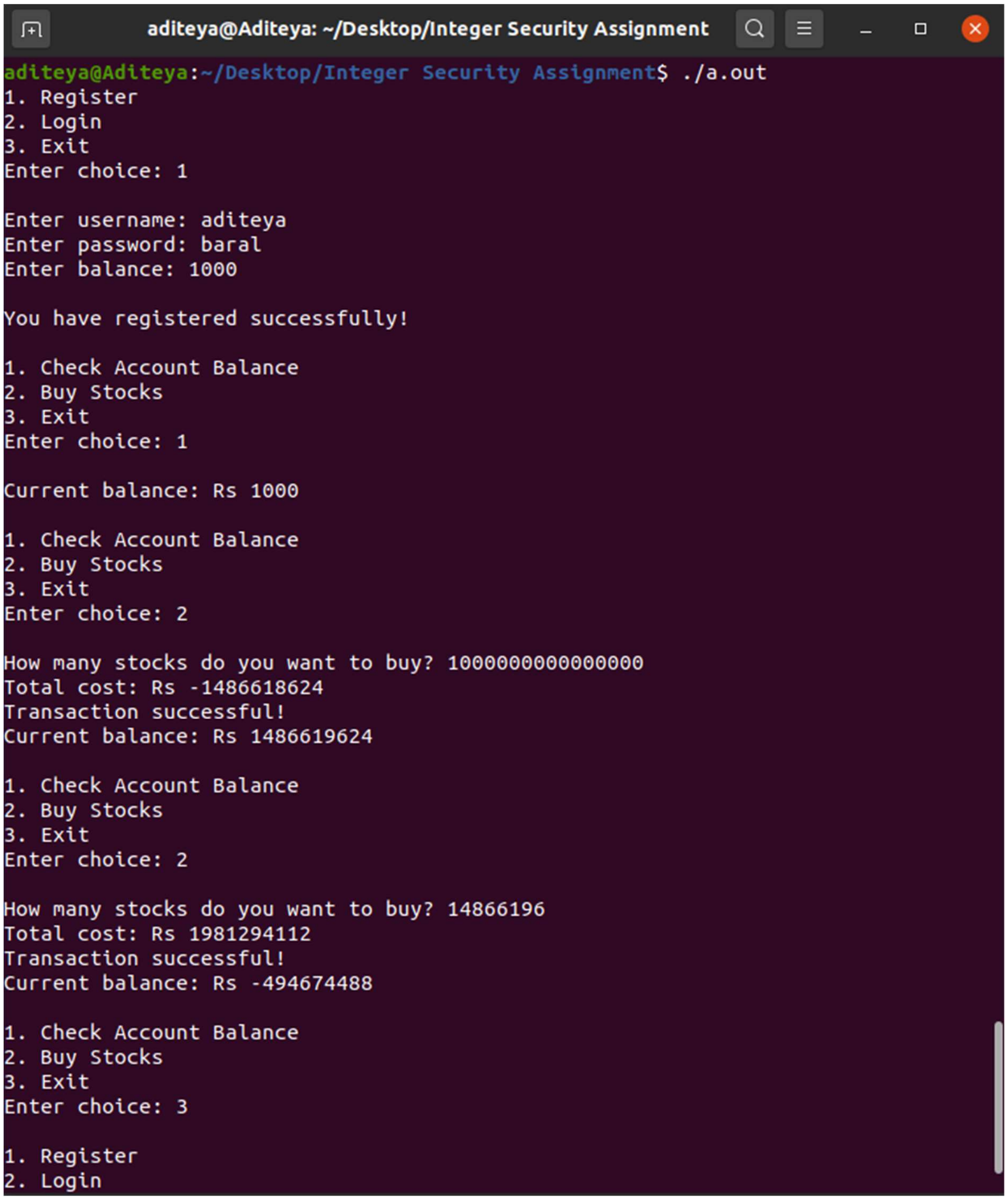
An integer value will overflow if we attempt to store a value larger than the maximum value it can store. It will cause it to loop the other way and take a value from the negative end. This can be shown by attempting to store a value larger than INT_MAX in an integer variable. Since there are no checks being performed here to ensure the validity of the numbers, we encounter an overflow.

In the below screenshot, we attempt to buy more than INT_MAX number of flags. This causes the number of flags to become negative, and hence the cost also takes a negative value. On subtracting this negative value from the current balance, this cost ends up getting added and we end up with 1000 plus the cost as the new balance.


Dependencies-vulnerability.c - Integer Security Assignment - Visual Studio Code
header.h
Main.c
Dependencies-vulnerability.c
Dependencies-secure.c
Dependencies-vulnerability.c > login()
adityea@Aditeya: ~/Desktop/Integer Security Assignment
Ln 83, Col 2 Spaces: 4 UTF-8 LF C Linux

```c
}
void activity()
{
    int choice,s,total;
    char ch;
    while(true)
    {
        printf("1. Check Account Balance\n2. Buy Stocks\n3. Exit\nEnter ch
        scanf("%d",&choice);
        printf("\n");
        if (choice==1)
            printf("Current balance: Rs %d\n",current->balance);
        else if (choice==2)
        {
            printf("How many stocks do you want to buy? ");
            scanf("%d",&s);
            total = s*1000;
            printf("Total cost: Rs %d\n",total);
            printf("Transaction successful!\n");
            current->stock+=1;
            current->balance-=total;
            printf("Current balance: Rs %d\n",current->balance);
            update();
        }
        else if (choice==3)
        {
            free(accounts);
            main();
        }
        else
            printf("\nInvalid choice! Try again.\n");
        printf("\n");
    }
}
void update()
{
    FILE* fp = fopen("Database.dat","w");
    for(int i=0;i<N;i++)
        fwrite(&accounts[i],sizeof(ACCOUNT),1,fp);
    fclose(fp);
}
```

```
aditeya@Aditeya:~/Desktop/Integer Security Assignment$ ./a.out
1. Register
2. Login
3. Exit
Enter choice: 1

Enter username: aditeya
Enter password: baral
Enter balance: 1000

You have registered successfully!

1. Check Account Balance
2. Buy Stocks
3. Exit
Enter choice: 1

Current balance: Rs 1000

1. Check Account Balance
2. Buy Stocks
3. Exit
Enter choice: 2

How many stocks do you want to buy? 1000000000000000
Total cost: Rs -1486618624
Transaction successful!
Current balance: Rs 1486619624

1. Check Account Balance
2. Buy Stocks
3. Exit
Enter choice: 2

How many stocks do you want to buy? 14866196
Total cost: Rs 1981294112
Transaction successful!
Current balance: Rs -494674488

1. Check Account Balance
2. Buy Stocks
3. Exit
Enter choice: 3

1. Register
2. Login
```

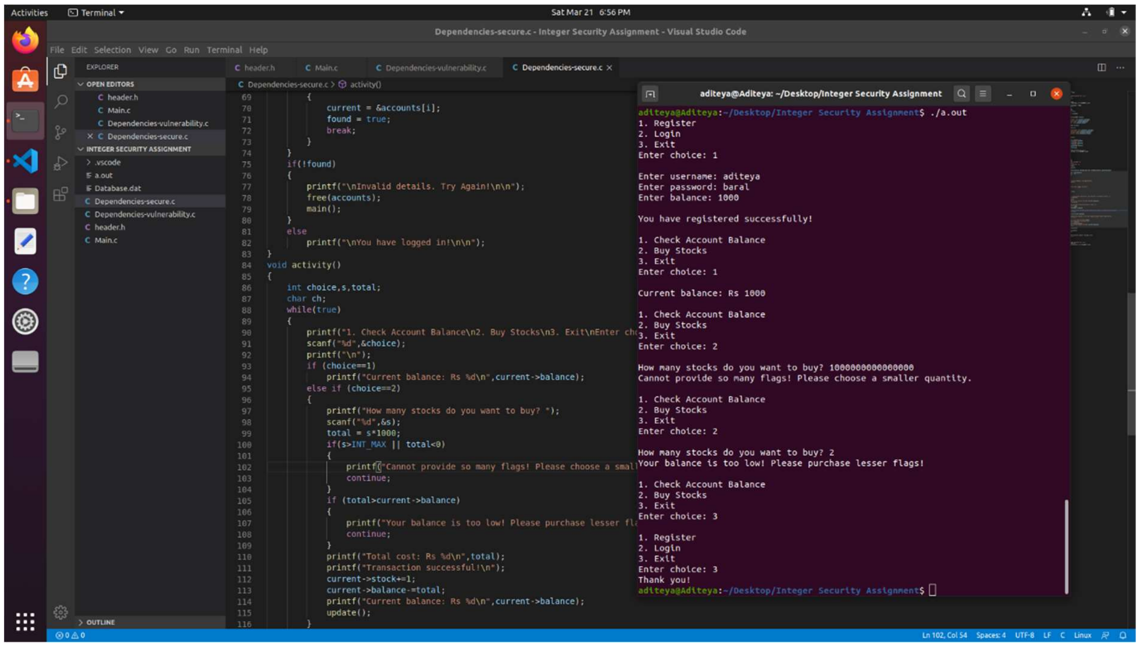


```
aditeya@Aditeya:~/Desktop/Integer Security Assignment$ ./a.out
1. Register
2. Login
3. Exit
Enter choice: 1

Enter username: aditeya
Enter password: baral
Enter balance: 1000

You have registered successfully!

1. Check Account Balance
2. Buy Stocks
3. Exit
Enter choice: 1

Current balance: Rs 1000

1. Check Account Balance
2. Buy Stocks
3. Exit
Enter choice: 2

How many stocks do you want to buy? 1000000000000000
Total cost: Rs -1486618624
Transaction successful!
Current balance: Rs 1486619624

1. Check Account Balance
2. Buy Stocks
3. Exit
Enter choice: 2

How many stocks do you want to buy? 14866196
Total cost: Rs 1981294112
Transaction successful!
Current balance: Rs -494674488

1. Check Account Balance
2. Buy Stocks
3. Exit
Enter choice: 3

1. Register
2. Login
```

No Vulnerability:

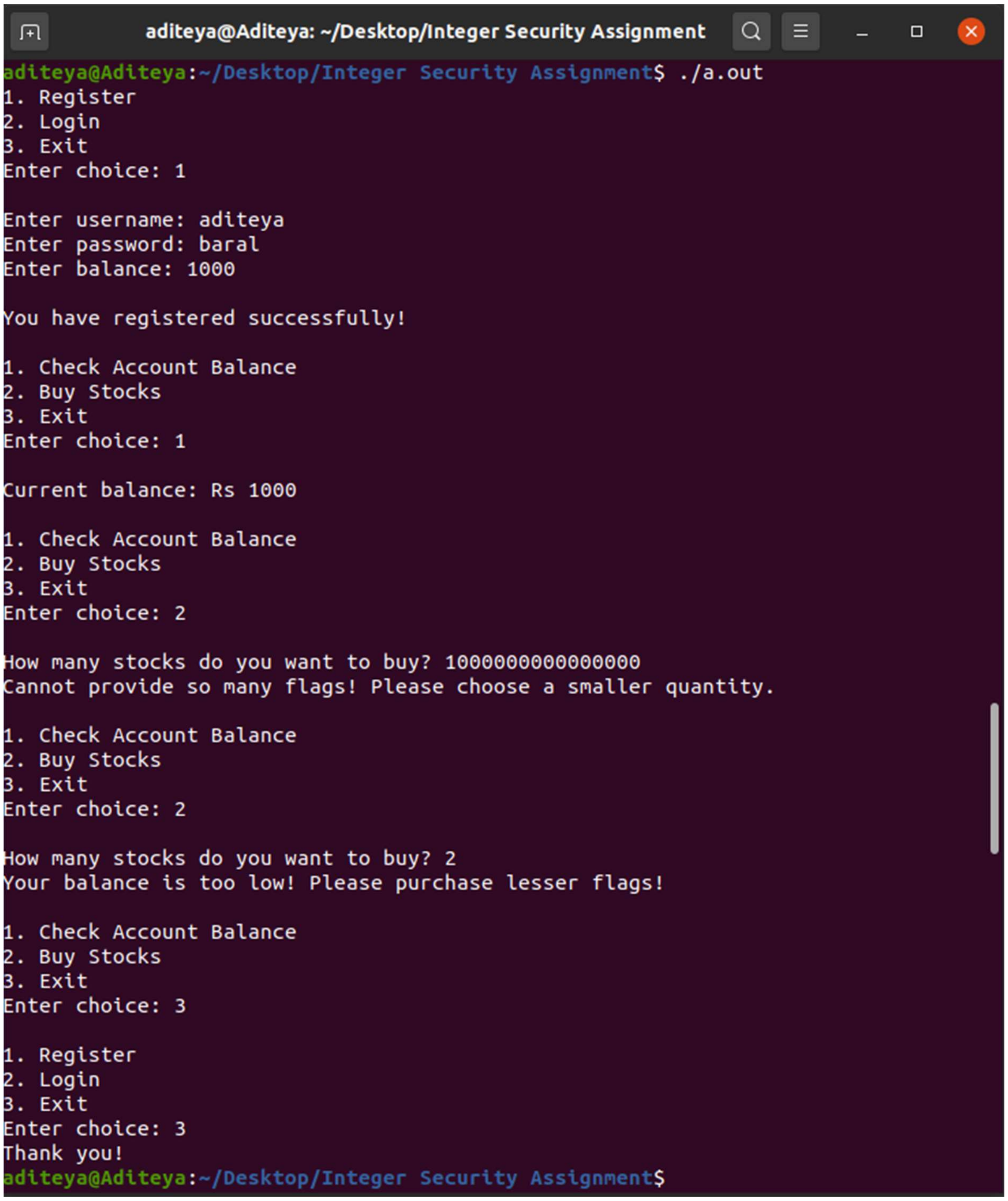
We can prevent such vulnerabilities by including small checks within our code that checks the value of an integer before performing other operations. To prevent an overflow here, we perform two checks. We first check if the number of desired flags is greater than INT_MAX, since this will definitely result in an overflow in the number of flags. However, it is still possible that the number of flags may be valid, but the total cost may overflow in value by exceeding INT_MAX, and hence it might take a negative value. Hence, we also check if the total cost is positive.

We also add in an extra condition to check if the cost of the flags is lesser than the available balance as a precautionary measure. Only if all these conditions are satisfied, we continue with the execution.



```c
            {
                current = &accounts[i];
                found = true;
                break;
            }
        }
        if(!found)
        {
            printf("\nInvalid details. Try Again!\n\n");
            free(accounts);
            main();
        }
        else
            printf("\nYou have logged in!\n\n");
}
void activity()
{
        int choice,s,total;
        char ch;
        while(true)
        {
            printf("1. Check Account Balance\n2. Buy Stocks\n3. Exit\nEnter ch
            scanf("%d",&choice);
            printf("\n");
            if (choice==1)
                printf("Current balance: Rs %d\n",current->balance);
            else if (choice==2)
            {
                printf("How many stocks do you want to buy? ");
                scanf("%d",&s);
                total = s*1000;
                if(s>INT_MAX || total<0)
                {
                    printf("Cannot provide so many flags! Please choose a smal
                    continue;
                }
                if (total>current->balance)
                {
                    printf("Your balance is too low! Please purchase lesser fl
                    continue;
                }
                printf("Total cost: Rs %d\n",total);
                printf("Transaction successful!\n");
                current->stock+=1;
                current->balance-=total;
                printf("Current balance: Rs %d\n",current->balance);
                update();
            }
```

```
aditeya@Aditeya:~/Desktop/Integer Security Assignment$ ./a.out
1. Register
2. Login
3. Exit
Enter choice: 1

Enter username: aditeya
Enter password: baral
Enter balance: 1000

You have registered successfully!

1. Check Account Balance
2. Buy Stocks
3. Exit
Enter choice: 1

Current balance: Rs 1000

1. Check Account Balance
2. Buy Stocks
3. Exit
Enter choice: 2

How many stocks do you want to buy? 1000000000000000
Cannot provide so many flags! Please choose a smaller quantity.

1. Check Account Balance
2. Buy Stocks
3. Exit
Enter choice: 2

How many stocks do you want to buy? 2
Your balance is too low! Please purchase lesser flags!

1. Check Account Balance
2. Buy Stocks
3. Exit
Enter choice: 3

1. Register
2. Login
3. Exit
Enter choice: 3
Thank you!
aditeya@Aditeya:~/Desktop/Integer Security Assignment$
```



aditeya@Aditeya: ~/Desktop/Integer Security Assignment

```
aditeya@Aditeya:~/Desktop/Integer Security Assignment$ ./a.out
1. Register
2. Login
3. Exit
Enter choice: 1

Enter username: aditeya
Enter password: baral
Enter balance: 1000

You have registered successfully!

1. Check Account Balance
2. Buy Stocks
3. Exit
Enter choice: 1

Current balance: Rs 1000

1. Check Account Balance
2. Buy Stocks
3. Exit
Enter choice: 2

How many stocks do you want to buy? 1000000000000000
Cannot provide so many flags! Please choose a smaller quantity.

1. Check Account Balance
2. Buy Stocks
3. Exit
Enter choice: 2

How many stocks do you want to buy? 2
Your balance is too low! Please purchase lesser flags!

1. Check Account Balance
2. Buy Stocks
3. Exit
Enter choice: 3

1. Register
2. Login
3. Exit
Enter choice: 3
Thank you!
aditeya@Aditeya:~/Desktop/Integer Security Assignment$
```