## B.Tech Program First Year Course: Experiential Learning Course Code: DA1001

**“Money-Splitter-Spili”**

by

## ADITYA GUPTA

**(Reg. No:219302313 )**

Under the guidance of

## MR DEVI PRASAD SHARMA

## Professor

INFORMATION TECHNOLOGY, Manipal University Jaipur, Jaipur

## Department of INFORMATION TECHNOLOGY Faculty of Engineering

**Manipal University Jaipur, India**

JULY, 2022

# Certificate

This is to certify that the project titled **“Money-Splitter”** is a record of the bona fide work done by **ADITYA GUPTA** (Reg No:219302313) submitted for the partial fulfilment of the requirements for the completion of the Experiential Learning (DA1001) course in the Department of Information Technology of Manipal University Jaipur, during the academic session March-July 2022.

|  |
| --- |
| *Signature of the mentor* |
| Mr. DEVI PRASAD SHARMA  Department of information technology |
| *Signature of the HoD* |
| Dr PANKAJ VYAS  Head of the Department Department of information technology |

# ACKNOWLEDGEMENT

In the accomplishment of this project successfully, many people have bestowed upon me their blessings and the heart pledged support, this time I am utilizing to thank all the people who have been concerned with this project.

Primarily I would thank god for being able to this project with success. Then I

would like to thank my Head of Department, Dr. Pankaj Vyas and my mentor Mr. Devi Prasad Sharma, whose valuable guidance has been the ones that helped me patch this project and make it full proof success. His suggestions and his instructions have served as the major contributor towards completion of the project.

Then I would like to thank my parents and friends who have helped me with their valuable suggestions and guidance has been very helpful in various phases of the completion of the project. Last but not the least I would like to thank my classmates who have helped me a lot.

## ADITYA GUPTA

(Reg. No: 219302313)

# Software & Hardware Requirements

# 

To become a programmer is to write and execute lots of code. To do that, you will need two different pieces of software: a **compiler** and a **text editor**. Below is a list of some popular C compilers. Note that some of the compilers listed below come as part of an [**integrated development environment**](http://wikipapers.referata.com/wiki/Integrated_development_environment) **(IDE).**

|  |  |
| --- | --- |
| **Name** | **Platform** |
| [Microsoft Visual Studio](https://en.wikipedia.org/wiki/Microsoft_Visual_Studio#Community) [Community](https://en.wikipedia.org/wiki/Microsoft_Visual_Studio#Community) | Windows |
| [Xcode](https://en.wikipedia.org/wiki/Xcode) | macOS, OSX |
| [Tiny C Compiler (TCC)](https://en.wikipedia.org/wiki/Tiny_C_Compiler) | GNU/Linux, Windows |
| [Clang](https://en.wikipedia.org/wiki/Clang) | GNU/Linux, Windows, Unix, OS X |
| [GNU C Compiler](https://en.wikipedia.org/wiki/GNU_Compiler_Collection) | GNU/Linux, [MinGW](http://mingw.org/) or [mingw-w64](https://mingw-w64.org/) (Windows), Unix, OS X. |

On **Microsoft Windows**, Dev-C++ is recommended for beginners because it is easy to use, free, and simple to install. Although the initial developer (Bloodshed) hasn’t updated it since 2005, a new version appeared in 2011, made by an independent programmer, and is being actively developed.[[2]](https://en.wikibooks.org/wiki/C_Programming/What_you_need_before_you_can_learn#cite_note-2) An alternate option for those working only in the Windows environment is the proprietary Microsoft Visual Studio Community which is free of charge and has an excellent debugger.On **Mac OS X**, the Xcode IDE provides the compilers needed to compile various source files. The newer versions do not include the command line tools. They need to be downloaded via Xcode->Preferences->Downloads.

# Introduction

More often, we plan trips with our friends and family members. Its fun, right? But the aftermath of dividing the expenses incurred in the trip might be tiring, and ruin the whole fun, isn’t it ? As a developer, I continuously try to develop things which can makes life much easier, something which people needs and can use in their daily life while making sure that it’s fun to develop it as well.

# This idea struck in my mind when I recently went on to a trip with my college buddies which was fun but the aftermath of expenses calculations were brainstorming and tiring. So, I thought can we develop a software which can fill this void, and then I thought to develop Spili. Spili is a software which can take list of expenses, divides them equally and outputs the final amount to pay for each individual. The central idea is “****How much money one needs to pay and to whom?****” Currently, I have tried it in C++ command line interface and planning to make a GUI of it as well. So, it works something like this :

# 

# Problem Statement:

Input :

* Name of all persons in the trip.
* List of expenses for each event in the trip, followed by name of the person who paid it.

Output :

* How much money one needs to pay and to whom?
* Example — person 1 has to pay Rs. 1000 to person 2 and so-on.

**Example**

Suppose there are three person — Mayank , Vivek and Spandan. They go on a trip and the expenses are distributed in the following way :-

***Cab****— Rs 200 paid by*Mayank.***Lunch****— Rs.1000 paid by*Spandan***Movie****— Rs.1500 paid by*Mayank***Cab****— Rs.300 paid by*Vivek.

So, the total amount spent in the trip by-

Mayank*—* Rs.(200+1500)=Rs.1700

Spandan *—* Rs.1000

Vivek *—* Rs.300

And the final adjustment of expenses, and the output should be :

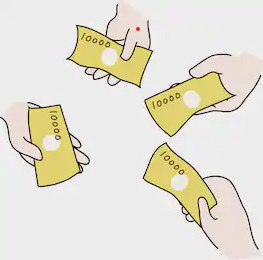
Spandan needs to pay Rs. 233 to Mayank

Vivek needs to pay Rs 466 to Mayank.

Vivek needs to pay Rs 233 to spandan.

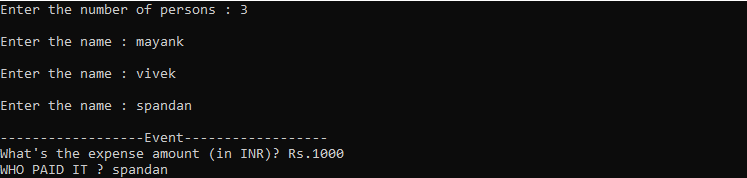
**Mayank need not to pay anyone , he will get Rs. 699 from his friends.**

**Spandan needs to pay Rs. 233 to Mayank.Vivek needs to pay Rs.466 to Mayank, Rs.233 to Spandan to clear all his debts**.

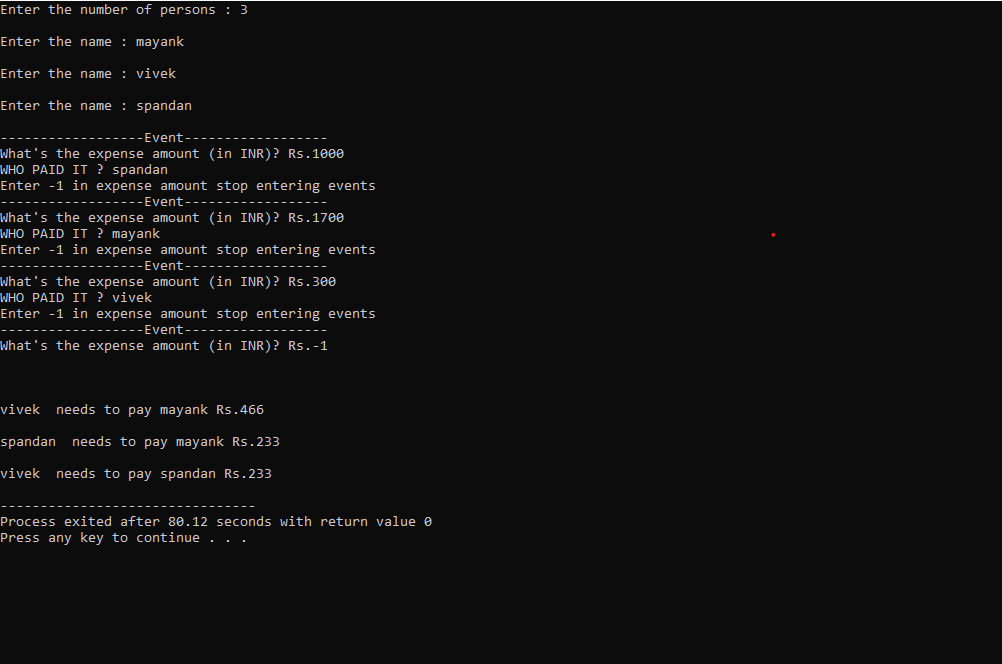


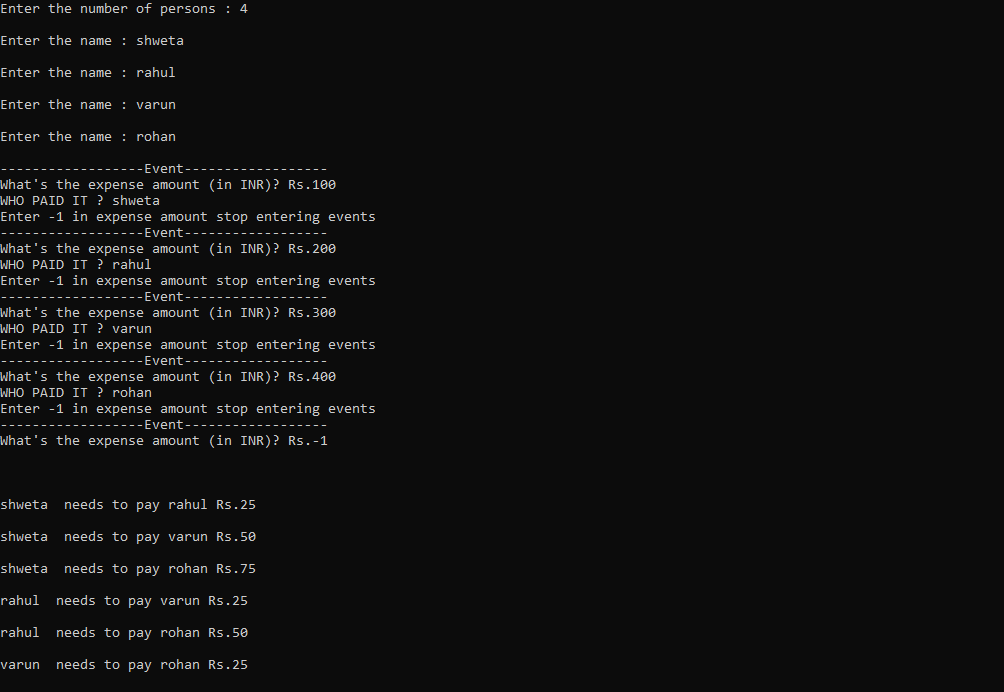
# Results and Discussions

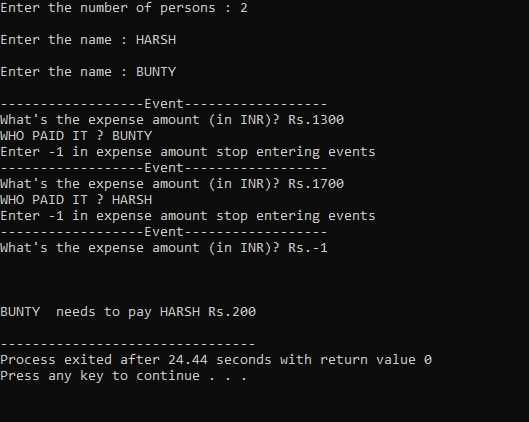
Take the number of persons in the trip. (say, **n**).



Take input a pair of <expense amount, name of person who paid it>



**SOME MORE EXAMPLE**



# Future prospects

* Optimising the matrix traversal, if possible.
* Develop GUI for this application.
* Make this system more robust by covering more number of corner test-cases.

# References

1. https://github.com/
2. https://www.geeksforgeeks.org/
3. https://www.codechef.com/
4. https://stackoverflow.com/
5. https://en.wikipedia.org/wiki/Wikipedia
6. <https://www.google.com/>
7. Yashavant Kanetkar - Let Us C-BPB Publications (2016)