

Planning Poker

Requested files: user.c (Download)

Type of work: Individual work

Grade settings: Maximum grade: 10

Dissable external file upload, paste and drop external content: Yes

Run: Yes **Evaluate:** Yes

Automatic grade: Yes

Problema:

Em engenharia de software, o *Planning Poker* consiste em que todos os membros da equipe de desenvolvimento (programadores, testadores, design e analistas) participem colocando a sua visão de complexidade em tempo e esforço do desenvolvimento do software utilizando um baralho com cartas numeradas em uma sequência similar a encontrada nos números de *Fibonacci*. O valor de uma carta $f(n)$ é calculado a partir da soma dos valores das duas cartas anteriores $f(n-1) + f(n-2)$.

O número de cartas que cada integrante tem a sua disposição em seu baralho é igual ao número " n " de membros da equipe. Cada baralho sempre tem uma carta 0 que é usada para indicar que o participante não está apto a estimar e o valor mínimo " v " da carta seguinte é definido pelo chefe do projeto de software. No há cartas repetidas no baralho, assim a carta seguinte tem uma xícara de café para o participante fazer uma sugestão de pausa (usaremos x para representar a xícara de café).

Dessa forma, se no equipe de desenvolvimento há dez membros ($n=10$) e o chefe de projeto define o valor mínimo de estimação como dois ($v=2$) então o baralho é composto das seguintes cartas:

0 2 x 4 6 10 16 26 42 68

Escreva um programa recursivo que dado um número n e v imprima o baralho para o planning poker.

Entrada e Saída:

A entrada será constituída por dois números inteiros n e v , onde o valor n é positivo maiores que 2 e v é positivo maior que 0 separados por um espaço em branco. Como saída você deve imprimir na tela do computador o baralho para o planning poker com seus valores separados por um espaço em branco.

Exemplos de Entrada	Exemplos de Saída
5 6	0 x 6 12 18
6 7	0 x 7 14 21 35
10 2	0 x 2 4 6 10 16 26 42 68

Requested files

user.c

```
1 #include<stdio.h>
2
3 int main() {
4     int n, v;
5     scanf("%d", &n, &v);
6     // escreva seu código aqui
7     return 0;
8 }
```

Execution files

vpl_run.sh

```
1 #!/bin/bash
2 # $Id: c_run.sh,v 1.3 2012-07-25 19:02:20 juanca Exp $
3 # Default C language run script for VPL
4 # Copyright (C) 2014 Juan Carlos Rodriguez-del-Pino
5 # License http://www.gnu.org/copyleft/gpl.html GNU GPL v3 or later
6 # Author Juan Carlos Rodriguez-del-Pino <jcrodriguez@dis.ulpgc.es>
7
8 #load common script and check programs
9 . common_script.sh
10 check_program gcc
11
12 #compile
13 SOURCE_FILES="user.c"
14 eval gcc -fno-diagnostics-color -o vpl_execution -std=c99 $SOURCE_FILES -lm -lutil
```

vpl_debug.sh

vpl_evaluate.sh