

UNIVERSITETI I PRISHTINËS
“HASAN PRISHTINA”
FAKULTETI I INXHINIERISË ELEKTRIKE DHE KOMPJUTERIKE



LËNDA: ARKITEKTURA E KOMPJUTERËVE
DETYRA: KODIMI NË MIPS ASSEMBLY LANGUAGE

Mësimdhënësit:

Dr. Valon Raca

MSc. Vlera Alimehaj

Studenti:

Fortesa Mujaj

ID: 190714100155

HYRJE

Opsioni A: Ky opsion ju ofron mundësinë që të notoheni maksimalisht 10%

Të shkruhet në MIPS assembly code kodi i mëposhtëm në C++:

```
#include <iostream>
using namespace std;
int fib(int x) {
    if((x==1) || (x==0)) {
        return(x);
    }else {
        return(fib(x-1)+fib(x-2));
    }
}
int main() {
    int x , i=0;
    cout << "Enter the number of terms of series : ";
    cin >> x;
    cout << "\nFibonnaci Series : ";
    while(i < x) {
        cout << " " << fib(i);
        i++;
    }
    return 0;
}
```

Kodi i dhënë është kodi për serinë Fibonacci. Seria Fibonacci është seri e shoqëruar me numrat: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ...

Në terma matematikë, seria F_n e Fibonacci definohet nga relacioni i përsëritjes:

$$F_n = F_{n-1} + F_{n-2} \text{ dhe nga } F_0 = 0, F_1 = 1.$$

Ekzistojnë dy mënyra për të koduar një program për serinë e Fibonacci:

1. Me rekursion dhe,
2. Pa rekursion.

Kodi yne i dhënë paraqet një mënyrë të kodimit të serisë së Fibonacci me rekursion.

Funksioni *fib* merr si parameter një numër të plotë, nëse ky numër i plotë është 1 atëherë kthen 1, nëse ky numër i plotë është 0 atëherë kthen 0. Në të kundërtën, ky funksion thirret në mënyrë rekursive për vlerat $x - 1$ dhe $x - 2$, në fund rezultati që fitohet nga vlerat e fituara për *fib*($x - 1$) dhe *fib*($x - 2$) dhe mbledhjen në mes tyre kthehet.

Përmes unazës *while*, varësisht se sa e kemi dhënë numrin e termave të serisë, paraqiten keto terma, po ashtu numri *i* rritet për 1. Unaza *while* do të përfundoj kur numri *i* është më i madh se numri i termave të serisë së përcaktuar nga ne.

REALIZIMI I KODIT NË MIPS

```
.data
messageOne:      .ascii "Enter the number of terms of series : "
messageTwo:      .ascii "\nFibonnaci Series : "
messageThree:    .ascii " "

.text
.globl main

main:
    # Printimi i stringut: "Enter the number of terms of series : "
    li    $v0, 4          # System call code per print_str
    la    $a0, messageOne # Adresa e string-ut per printim
    syscall              # Printo string-un

    # Leximi i numrit x nga tastiera
    li    $v0, 5          # System call code per read_int
    syscall              # Vlera e x tashme gjendet ne $v0

    # Ruajtja e vleres se regjistrir $v0 ne $t1
    addi   $t1, $v0, 0     # x ruaje inputin ne regjistrin $t1

    # Printimi i stringut: "\nFibonnaci Series : "
    li    $v0, 4          # System call code per print_str
    la    $a0, messageTwo # Adresa e string-ut per printim
    syscall              # Printo string-un

    # Inicializimi i vleres se regjistrir $a0
    addi   $a0, $zero, 0   # $a0 --> i, i=0

    #-----While Loop-----#

loop:
    # Kushti if(i>=x) perfundo main funksionin
    bge    $a0, $t1, done  # Nese $a0 >= $t1 goto etiketa done

    # Alokimi i hapsires ne stack
    addi   $sp, $sp, -4    # Alokimi i 4 bytes ne stack per i
    sw     $a0, 0($sp)     # E "push" i ne stack.
    jal    fibonnaci       # Thirret funksioni fibonnaci.

    # Printimi i returned value te funksionit fibonnaci
    addi   $a0, $v0, 0
    li    $v0, 1          # System call code per print_int
    syscall              # Printo integer-in
```

```

# Printimi i stringut: " "
li    $v0, 4           # System call code per print_str
la    $a0, messageThree # Adresa e string-ut per printim
syscall               # Printo string-un

# Dealokimi i hapsires se rezervuar ne stack
lw    $a0, 0($sp)      # E "restore" $a0 origjinale
addi  $sp, $sp, 4      # E "pop" $a0 nga stack

# Inkrementimi i argumentit per 1
addi  $a0, $a0, 1      # i++

# Vazhdo perseri me unazen
j loop                # Jump to loop ose goto loop

done:
# Exit the main function
li    $v0, 10          # System call code per exit
syscall

#-----Fibonnaci Function-----#

fibonnaci:
# Kodi i dhene per kushtin (x==1) || (x==0)
beq   $a0, 1, return_one # Nese x==1 kthe 1, ose goto return_one
beqz  $a0, return_zero   # Nese x==0 kthe 0, ose goto return_zero

# Alokimi i hapsires se nevojshme ne memorie
addi  $sp, $sp, -12      # Alokimi i hapsires ne stack per 3 integera
sw    $ra, 8($sp)        # $ra -> $sp + 8
sw    $a0, 4($sp)        # $a0 -> $sp + 4

# fib(x-1) + fib (x-2)
addi  $a0, $a0, -1      # x-1
jal   fibonnaci          # fib(x-1)
sw    $v0, 0($sp)        # fib(x-1) -> $sp + 0

lw    $a0, 4($sp)        # E "restore" parametrin origjinal
addi  $a0, $a0, -2      # x-2
jal   fibonnaci          # fib(x-2)

lw    $t0, 0($sp)        # fib(x-1) -> $t0
add  $v0, $t0, $v0      # fib(x-1) + fib(x-2) -> $v0

#Dealokimi i hapsires se rezervuar ne stack
lw    $ra, 8($sp)        # E "restore" return adresen
addi  $sp, $sp, 12      # Dealokon hapsiren ne stack
jr    $ra               # Kce te return address

```

```

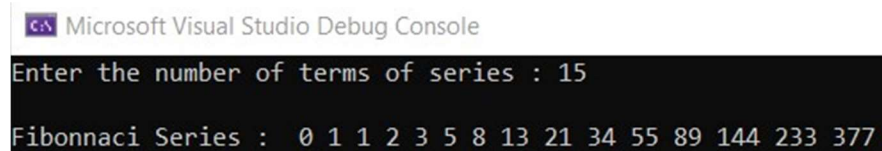
return_one:
    #Kthimi i vleres 1
    addi $v0, $zero, 1      # Vlera 1 vendoset ne $v0
    jr   $ra                # Kce te return address

return_zero:
    #Kthimi i vleres 0
    addi $v0, $zero, 0      # Vlera 0 vendoset ne $v0
    jr   $ra                # Kce te return address

```

TESTIMET NË QTSPIM

Kodi në C++ ku numri i termave të serisë është 15:

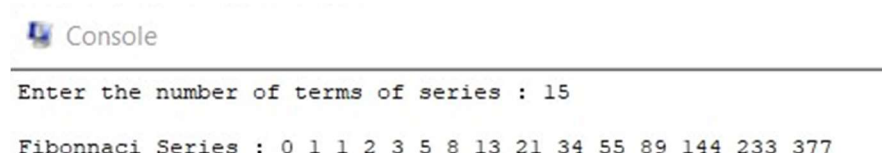


```

Microsoft Visual Studio Debug Console
Enter the number of terms of series : 15
Fibonnaci Series : 0 1 1 2 3 5 8 13 21 34 55 89 144 233 377

```

Kodi në MIPS ku numri i termave të serisë është 15:

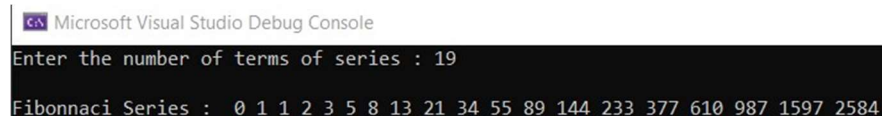


```

Console
Enter the number of terms of series : 15
Fibonnaci Series : 0 1 1 2 3 5 8 13 21 34 55 89 144 233 377

```

Kodi në MIPS ku numri i termave të serisë është 19:

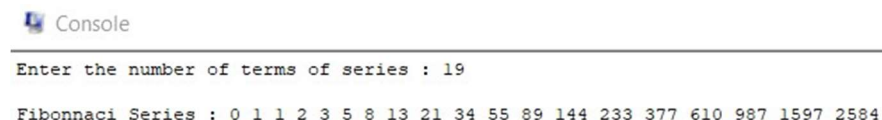


```

Microsoft Visual Studio Debug Console
Enter the number of terms of series : 19
Fibonnaci Series : 0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584

```

Kodi në MIPS ku numri i termave të serisë është 19:



```

Console
Enter the number of terms of series : 19
Fibonnaci Series : 0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584

```

PËRFUNDIMI

Si përfundim, gjatë zhvillimit të detyrës mesova shumë jo vetëm se si kodohen dhe përkthehen programet e ndryshme nga nje gjuhë sic është C++ në nje gjuhë më të ulët sic është MIPS Assembly, mirëpo edhe se si punon memoria dhe procesori gjatë ekzekutimit të një programi të tillë. Përvec kësaj, fitova edhe njohuri për Serinë Fibonacci, e cila është tëjet e rëndësishme në natyrë për shkak se luan një rol jetik në phyllotaxis, i cili studion rregullimin e gjetheve, degëve, luleve ose farave në bimë, me qëllimin kryesor për të theksuar modelin e rregullt te ekzistencës.