Contents

[Assembly Language Test Program 2](#_Toc4853902)

[1. Dynamic Memory Allocation and Freeing 2](#_Toc4853903)

[2. Using Stack 2](#_Toc4853904)

[Machine Language Test Program 3](#_Toc4853905)

[1. Dynamic Memory Allocation and Freeing 3](#_Toc4853906)

[2. Using Stack 3](#_Toc4853907)

[1. Symbol Table 3](#_Toc4853908)

# Assembly Language Test Program

## Dynamic Memory Allocation and Freeing

## Using Stack

|  |  |  |  |
| --- | --- | --- | --- |
| **Label** | **Mnemonic** | **Operands** | **Description** |
| main | Function |  | Start of main function |
| Start | Move | R2, 10 | Set R2 to 10, memory size to be requested |
|  | SystemCall | 4 | Request OS to allocate dynamic memory |
|  | Move | R3, R1 | Set R3 to value in R1, dynamic memory address |
|  | Move | R4, 10 | Set R4 to 10, loop iterations |
| LoopPush | Push | 1980 | Push 1980 onto stack |
|  | Subtract | R4, 1 | Subtract 1 from R4, R4 = R4 – 1 |
|  | BrOnPlus | R4, LoopPush | Branch to LoopPush if R4 > 0 |
|  | Move | R4, 10 | Set R4 to 10, loop iterations |
| LoopPop | Pop | (R3)++ | Pop 1980 into address in R3 and increment R3 |
|  | Subtract | R4, 1 | Subtract 1 from R4, R4 = R4 – 1 |
|  | BrOnPlus | R4, LoopPop | Branch to LoopPop if R4 > 0 |
|  | SystemCall | 4 | Request OS to release (free) allocated memory |
|  | Halt |  | Stop execution of program |
|  | End | Start | Execution starts at Move instruction |

# Machine Language Test Program

## Dynamic Memory Allocation and Freeing

## Using Stack

**Address Content Comment**

36 51260 // Start Move R2, 10; set R2 to 10

37 10 // Immediate operand value of 10

38 120000 // SystemCall 4; make system call with ID 4

39 4 // System call ID

40 51311 // Move R3, R1; set R3 to value in R1

41 51460 // Move R4, 10; set R4 to 10

42 10 // Immediate operand value of 10

43 106000 // LoopPush Push 1980; push 1980 onto stack

44 1980 // Immediate operand value of 1980

45 21460 // Subtract R4, 1; subtract 1 from R4

46 1 // Immediate operand value of 1

47 81400 // BrOnPlus LoopPush; branch on plus to LoopPush

48 43 // Address of LoopPush

49 51460 // Move R4, 10; set R4 to 10

50 10 // Immediate operand value of 10

51 113500 // LoopPop Pop (R3)++; op 1980 into address in R3 and increment R3

52 21460 // Subtract R4, 1; subtract 1 from R4

53 1 // Immediate operand value of 1

54 81400 // BrOnPlus LoopPop; branch on plus to LoopPop

55 51 // Address of LoopPop

56 120000 // SystemCall 5; make system call with ID 5

57 5 // System call ID

58 0 // HALT

-1 36 // End of program; PC = 36

# Symbol Tables

## Dynamic Memory Allocation and Freeing

## Using Stack

|  |  |
| --- | --- |
| **Symbol** | **Value (Address)** |
| main | 0 |
| Evensum | 0 |
| Addend | 1 |
| Temp | 2 |
| Final | 3 |
| Start | 4 |
| Loop | 8 |
| Negative | 31 |