**Program #1: – Dynamic Memory Allocation and Freeing**

Assembly Program w/ Comments

**Label Instruction Operand Comment**

M Long 5 Long value M, constant value 5

R Long 145 Long value R, random value

Space Long 150 Long value to specify amount of space to allocate

Start Move R2,Space Put space into GPR 2

Move R3,R Move R into GPR3 for arithmetic

Move R4,M Move M into GPR4 for arithmetic

SystemCall 4 Use System Call – Memory allocation

Loop Add R4,R3 Start of Loop, Add value of R3 to R4

Subtract R2,1 Subtract 1 from R2

BranchOnZero R2,Next Once R2 reaches 0, go to Next label

Branch Loop Branch back to Loop Label

Next Move R2,100 Place 100 into R2

SystemCall 5 Use System Call – Free the memory

Halt Stop execution of program

End Start Execution begins at Start Label

Machine Code w/ Comments

**Address Content Comment**

0 5 M; Fixed value 150

1 145 R; Random value 145

2 150 Space; specified for memory allocation

3 51250 Start; Move R2, Space

4 2 Address of space

5 51350 Move R3, R Value

6 1 Address of R

7 51450 Move R4, M value

8 0 Address of M

9 126000 System call

10 4 Memory Allocation system call

11 11413 Loop; Add R4, R3

12 21260 Subtract R2, 1

13 1 immeadiate operand value of 100

14 91250 BranchOnZero, R2 (if R2 == 0)

15 18 Address of Next

16 65000 Branch to Loop

17 11 Address of Loop

18 51260 Next; Move R2, 100

19 100 Immeadiate Operand Value of 100

20 126000 System Call

21 5 Memory Free System Call

22 00000 Halt

-1 3 End of program indicator, Start address = 3

Symbol Table

|  |  |
| --- | --- |
| **Symbol** | **Value (Address)** |
| M | 0 |
| R | 1 |
| Space | 2 |
| Start | 3 |
| Loop | 11 |
| Next | 18 |

**Program #2: – Using Stack**

Assembly Program w/ Comments

**Label Instruction Operand Comment**

Mcounter Long 12 MCounter; set to 12

Start Branch Loop Start; Branch to Loop

Loop Subtract Mcounter,1 Loop; Subtract 1 from MCounter

One Push 4916 One; Push value 4916 onto stack

Two Push 9058 Two; Push value 9058 onto stack

Three Push 6434 Three; Push value 6364 onto stack

Four Push 8400 Four; Push value 8400 onto stack

Five Push 1860 Five; Push value 1860 onto stack

Six Push 9272 Six; Push value 9272 onto Stack

Seven Push 2054 Seven; Push value 2054 onto Stack

Eight Push 7896 Eight; Push value 7896 onto Stack

Nine Push 3358 Nine; Push value 3358 onto Stack

Ten Push 5768 Ten; Push value 5768 onto Stack

Pop Pop the top value of stack

Pop Pop the top value of stack

Pop Pop the top value of stack

Pop Pop the top value of stack

Pop Pop the top value of stack

Pop Pop the top value of stack

Pop Pop the top value of stack

Pop Pop the top value of stack

Pop Pop the top value of stack

Pop Pop the top value of stack

BranchOnPlus Mcounter,Loop If Mcounter > 0, go to Loop

Halt Stop Program execution

End Start Execution begins at Start Label

Machine Code w/ Comments

**Address Content Comment**

0 12 Mcounter; Value set to 12

1 65000 Start; Branch to Loop

2 3 Address of Loop

3 25060 Loop; Subtract M. M = M - 1

4 0 Address of M

5 1 Immeadiate Operand value of 1

6 106000 Push Value onto stack

7 4916 Immeadiate Operand value of 4916

8 106000 Push Value onto stack

9 9058 Immeadiate Operand value of 9058

10 106000 Push Value onto stack

11 6434 Immeadiate Operand value of 6364

12 106000 Push Value onto stack

13 8400 Immeadiate Operand value of 8400

14 106000 Push Value onto stack

15 1860 Immeadiate Operand value of 1860

16 106000 Push Value onto stack

17 9272 Immeadiate Operand value of 9272

18 106000 Push Value onto stack

19 2054 Immeadiate Operand value of 2054

20 106000 Push Value onto stack

21 7896 Immeadiate Operand value of 7896

22 106000 Push Value onto stack

23 3358 Immeadiate Operand value of 3358

24 106000 Push Value onto stack

25 5768 Immeadiate Operand value of 5786

26 110000 Pop the top value of the stack

27 110000 Pop the top value of the stack

28 110000 Pop the top value of the stack

29 110000 Pop the top value of the stack

30 110000 Pop the top value of the stack

31 110000 Pop the top value of the stack

32 110000 Pop the top value of the stack

33 110000 Pop the top value of the stack

34 110000 Pop the top value of the stack

35 110000 Pop the top value of the stack

36 85050 Branch on Plus, if MCounter > 0 go to loop

37 0 Address of Mcounter

38 3 Address for Loop Label

39 00000 Halt

-1 1 End of program indicator, Start address = 1

Symbol Table

|  |  |
| --- | --- |
| **Symbol** | **Value (Address)** |
| MCounter | 0 |
| Start | 1 |
| Loop | 3 |
| One | 7 |
| Two | 9 |
| Three | 11 |
| Four | 13 |
| Five | 15 |
| Six | 17 |
| Seven | 19 |
| Eight | 21 |
| Nine | 23 |
| Ten | 25 |
|  |  |
|  |  |

**Program #3: – Performing Input and Output Operations**

Assembly Program w/ Comments

**Label Mnemonic Operands Description**

main Function Start of main function

Count1 Long 5 Initialize loop1 counter to 5.

Count2 Long 5 Initialize loop2 counter to 5.

Start Move R2,9 Initialize R2 to 9.

SysC1 SystemCall 4 Systemcall 4 - MemAlloc // Allocate Memory

Loop1 SystemCall 8 Systemcall 8 - IO\_GetC // read character

Subtract Count1,1

BrOnPlus Loop1

Loop2 SystemCall 9 Systemcall 9 - IO\_PutC // Print character

Subtract Count2,1

BrOnPlus Loop2

SysC2 SystemCall 5 Systemcall 5 - MemFree // Free/release memory

End Halt Stop execution of the program.

End Start Execution starts at the Move instruction

Machine Code w/ Comments

**Address Content Comment**

0 5 Count1; value of 5

1 5 Count2; value of 5

2 51260 Start; Move R2, 9

3 9 immeadiate operand value of 9

4 126000 SysC1; SystemCall

5 4 Memory Allocation system call

6 126000 Loop1; System Call

7 8 IO\_GetC system call

8 25060 Subtract Count1, 1

9 0 address of Count1

10 1 Immeadiate Operand value of 1

11 85050 If Count1 > 0 Branch to Loop1

12 0 Address of Count1

13 6 Address of Loop1 Label

14 126000 Loop2; System Call

15 9 IO\_PutC

16 25060 Subtract Count2, 1

17 1 Address of Count2

18 1 immeadiate operand of 1

19 85050 If Count 2 > 0 goto Loop2

20 1 Address of Count1

21 14 Address of Loop2

22 126000 SysC2; System Call

23 5 Memory Free System call

24 00000 End; Halt

-1 2 End of Program indicator, program starts at 2

Symbol Table

|  |  |
| --- | --- |
| **Label** | **Value (Address)** |
| Count1 | 0 |
| Count2 | 1 |
| Start | 2 |
| SysC1 | 4 |
| Loop1 | 6 |
| Loop2 | 14 |
| SysC2 | 22 |
| End | 24 |