

CMPE 220: System Software
Assignment 4: Procedures and Function Calls

By

Hitesh Padekar (009273303)
Aarohi Oza (010118654)
Venkata Sowmya Paku (010041291)
Vishnu Vardhana Reddy Mandalapu (010013341)

1. C function for Binary Search using recursion:

```
/*
 * Function: binary_recursion
 * Parameters:
 * Input:
 *      array : Starting location of the array of elements
 *      size : Size of the array
 *      key : Element or value to be searched in the array
 *      lower: Lower bound of the array to be searched
 *      upper: Upper bound of the array to be searched
 * Output:
 *      returns an integer value representing location of the element to be
 *      searched in an array.
 */

int binary_recursion(int array[], int size, int key, int lower, int upper){

    int mid, flag=0;

    if(lower <= upper){
        mid=(lower + upper) / 2;
        if(key == array[mid]){
            flag = 1;
        }
        else if(key < array[mid]){
            return binary_recursion(array, size, key, lower, mid-1);
        }
        else
            return binary_recursion(array, size, key, mid+1, upper);
    }
    else
        return mid;
}
```

Corresponding assembly code:

```
BINARY:
cmp R3, R2
ja FINISH
mov R2, R7
mov R3, R8
add R7, R8
shr 1, R8
cmpv [R8], R4
jz SUCCESS
jae HIGH
```

```
LOW:
subi 1, R8
mov R8, R3
jmp BINARY
```

```
HIGH:
addi 1, R8
mov R8, R2
jmp BINARY
```

```
FINISH:
cmpv [R8], R4
jz SUCCESS
jmp FAIL
```

```
SUCCESS:
mov R8, R0
addi 1, R0
push R0
jmp EXIT
```

```
FAIL:
mul 0, R0
push R0
```

```
EXIT:
pop R0
```

Equivalence of assembly code to C code:

Parameter	C Code and Variables	Assembly Code and Variables
Array index	int array[]	Starts from memory address 0
Array Size	int size	Register R3 at the start, later R3 is used for upper bound
Upper bound of the array in search	int upper	Register R3
Lower bound of the array in search	int lower	Register R2
Key value to be searched	int key	Register R4
Mid element of the array	int mid	Register R8
Result	int mid	Register R0
Calculate mid value	mid=(lower + upper) / 2;	<pre> mov R2, R7 mov R3, R8 add R7, R8 shr 1, R8 </pre>
Recursive Call if key < array[mid]	return binary(array, size, key, lower, mid-1);	<pre> LOW: subi 1, R8 mov R8, R3 jmp BINARY </pre>
Recursive Call if key > array[mid]	return binary(array, size, key, mid+1, upper);	<pre> HIGH: addi 1, R8 mov R8, R2 jmp BINARY </pre>
Return result	<pre> return mid; or return values of the recursive calls </pre>	<pre> SUCCESS: mov R8, R0 addi 1, R0 push R0 jmp EXIT FAIL: mul 0, R0 push R0 EXIT: pop R0 </pre>

Instructions for executing the program:

1. Compiling the program:

`gcc best_cpu_4.c -o best_cpu`

2. Executing the program:

- a. Update "instructions.txt" file and place it in the same directory as the best_cpu executable.

- b. Update numbers in ascending order in "numbers.txt" file.

- i. This file contains integer numbers separated by space " " character.
 - ii. The first number in the file represents the number to be searched and subsequent numbers indicate the numbers in the array element.

e.g. if the contents of "numbers.txt" file are:

130 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170

Then,

130 = the key element to be searched

numbers 10 to 170 are the array elements

- c. To disable display of the memory contents for the best_cpu program optionally we can provide "-NoMemDisplay" option while running the executable through command line.
- d. To run the executable type in shell:
`./best_cpu -NoMemDisplay`

Program output and screenshots:

For each of the case if the the element is found in the array then Register value R1 contains the location / index of the array (starting from index 1) if not found then R1 value will be 0. R4 contains the key element to be searched.

Case 1: Array has even or odd number of elements

Set 1: Odd number of elements

Key = 130

10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170

```
*****
CPU State After execution of instruction
*****

*****
General Purpose Registers
*****
R0 = 13      R1 = 0      R2 = 12      R3 = 12
R4 = 130     R5 = 0      R6 = 0      R7 = 11
R8 = 12      R9 = 0      R10 = 0     R11 = 0
R12 = 0     R13 = 0     R14 = 0     R15 = 0

*****
Program Counter
*****
PC = 232

*****
Stack Pointer
*****
SP = 1024

*****
Base Pointer and Other
*****
BP = 500      AX = 0    BX = 0    CX = 0

*****
Flags
*****
flag = 0
CF = 0  PF = 0  AF = 0  ZF = 0
SF = 0  TF = 0  IF = 0  DF = 0  OF = 0

hitesh@master:~/220/lab4$
```

Set 2: Odd number of elements

Key = 20

10 20 30

```
*****
CPU State After execution of instruction
*****

*****
General Purpose Registers
*****
R0 = 2      R1 = 0      R2 = 0      R3 = 3
R4 = 20     R5 = 0      R6 = 0      R7 = 0
R8 = 1      R9 = 0      R10 = 0     R11 = 0
R12 = 0     R13 = 0     R14 = 0     R15 = 0

*****
Program Counter
*****
PC = 232

*****
Stack Pointer
*****
SP = 1024

*****
Base Pointer and Other
*****
BP = 500     AX = 0    BX = 0    CX = 0

*****
Flags
*****
flag = 0
CF = 0  PF = 0  AF = 0  ZF = 0
SF = 0  TF = 0  IF = 0  DF = 0  OF = 0

hitesh@master:~/220/lab4$
```

Set 3: Odd number of elements

Key = 10

10 20 30 40 50 60 70 80 90

```
*****
CPU State After execution of instruction
*****

*****
General Purpose Registers
*****
R0 = 1      R1 = 0      R2 = 0      R3 = 0
R4 = 10     R5 = 0      R6 = 0      R7 = 0
R8 = 0      R9 = 0      R10 = 0     R11 = 0
R12 = 0     R13 = 0     R14 = 0     R15 = 0

*****
Program Counter
*****
PC = 232

*****
Stack Pointer
*****
SP = 1024

*****
Base Pointer and Other
*****
BP = 500     AX = 0    BX = 0    CX = 0

*****
Flags
*****
flag = 0
CF = 0  PF = 0  AF = 0  ZF = 0
SF = 0  TF = 0  IF = 0  DF = 0  OF = 0

hitesh@master:~/220/lab4$
```

Set 4: Even number of elements

Key = 160

10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160

```
*****
CPU State After execution of instruction
*****

*****
General Purpose Registers
*****
R0 = 16      R1 = 0      R2 = 15      R3 = 16
R4 = 160     R5 = 0      R6 = 0      R7 = 15
R8 = 15      R9 = 0      R10 = 0     R11 = 0
R12 = 0     R13 = 0     R14 = 0     R15 = 0

*****
Program Counter
*****
PC = 232

*****
Stack Pointer
*****
SP = 1024

*****
Base Pointer and Other
*****
BP = 500      AX = 0    BX = 0    CX = 0

*****
Flags
*****
flag = 0
CF = 0  PF = 0  AF = 0  ZF = 0
SF = 0  TF = 0  IF = 0  DF = 0  OF = 0

hitesh@master:~/220/lab4$
```


Set 5: Even number of elements

Key = 20

10 20

```
*****
CPU State After execution of instruction
*****

*****
General Purpose Registers
*****
R0 = 2      R1 = 0      R2 = 0      R3 = 2
R4 = 20     R5 = 0      R6 = 0      R7 = 0
R8 = 1      R9 = 0      R10 = 0     R11 = 0
R12 = 0     R13 = 0     R14 = 0     R15 = 0

*****
Program Counter
*****
PC = 232

*****
Stack Pointer
*****
SP = 1024

*****
Base Pointer and Other
*****
BP = 500     AX = 0    BX = 0    CX = 0

*****
Flags
*****
flag = 0
CF = 0  PF = 0  AF = 0  ZF = 0
SF = 0  TF = 0  IF = 0  DF = 0  OF = 0

hitesh@master:~/220/lab4$
```

Set 6: Even number of elements

Key = 30

10 20 30 40 50 60

```
*****
CPU State After execution of instruction
*****

*****
General Purpose Registers
*****
R0 = 3      R1 = 0      R2 = 2      R3 = 2
R4 = 30     R5 = 0      R6 = 0      R7 = 0
R8 = 2      R9 = 0      R10 = 0     R11 = 0
R12 = 0     R13 = 0     R14 = 0     R15 = 0

*****
Program Counter
*****
PC = 232

*****
Stack Pointer
*****
SP = 1024

*****
Base Pointer and Other
*****
BP = 500     AX = 0    BX = 0    CX = 0

*****
Flags
*****
flag = 0
CF = 0  PF = 0  AF = 0  ZF = 0
SF = 0  TF = 0  IF = 0  DF = 0  OF = 0

hitesh@master:~/220/lab4$
```

Case 2: Search on an element > middle of the array

Set 1: Key = 170

10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170

```
*****
CPU State After execution of instruction
*****

*****
General Purpose Registers
*****
R0 = 17      R1 = 0      R2 = 16      R3 = 17
R4 = 170     R5 = 0      R6 = 0      R7 = 16
R8 = 16      R9 = 0      R10 = 0     R11 = 0
R12 = 0     R13 = 0     R14 = 0     R15 = 0

*****
Program Counter
*****
PC = 232

*****
Stack Pointer
*****
SP = 1024

*****
Base Pointer and Other
*****
BP = 500     AX = 0    BX = 0    CX = 0

*****
Flags
*****
flag = 0
CF = 0  PF = 0  AF = 0  ZF = 0
SF = 0  TF = 0  IF = 0  DF = 0  OF = 0

hitesh@master:~/220/lab4$
```

Set 2: Key = 160

10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170

```
*****
CPU State After execution of instruction
*****

*****
General Purpose Registers
*****
R0 = 16      R1 = 0      R2 = 14      R3 = 17
R4 = 160     R5 = 0      R6 = 0      R7 = 14
R8 = 15      R9 = 0      R10 = 0     R11 = 0
R12 = 0      R13 = 0     R14 = 0     R15 = 0

*****
Program Counter
*****
PC = 232

*****
Stack Pointer
*****
SP = 1024

*****
Base Pointer and Other
*****
BP = 500     AX = 0     BX = 0     CX = 0

*****
Flags
*****
flag = 0
CF = 0  PF = 0  AF = 0  ZF = 0
SF = 0  TF = 0  IF = 0  DF = 0  OF = 0

hitesh@master:~/220/lab4$
```

Set 3: Key = 100

10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170

```
*****
CPU State After execution of instruction
*****

*****
General Purpose Registers
*****
R0 = 10      R1 = 0      R2 = 9      R3 = 9
R4 = 100     R5 = 0      R6 = 0      R7 = 9
R8 = 9       R9 = 0      R10 = 0     R11 = 0
R12 = 0      R13 = 0     R14 = 0     R15 = 0

*****
Program Counter
*****
PC = 232

*****
Stack Pointer
*****
SP = 1024

*****
Base Pointer and Other
*****
BP = 500     AX = 0    BX = 0    CX = 0

*****
Flags
*****
flag = 0
CF = 0  PF = 0  AF = 0  ZF = 0
SF = 0  TF = 0  IF = 0  DF = 0  OF = 0

hitesh@master:~/220/lab4$
```

Set 4: Key = 160

90 100 110 120 130 140 150 160

```
*****
      CPU State After execution of instruction
*****

*****
      General Purpose Registers
*****
R0 = 8      R1 = 0      R2 = 7      R3 = 8
R4 = 160     R5 = 0      R6 = 0      R7 = 7
R8 = 7      R9 = 0      R10 = 0     R11 = 0
R12 = 0     R13 = 0     R14 = 0     R15 = 0

*****
      Program Counter
*****
PC = 232

*****
      Stack Pointer
*****
SP = 1024

*****
      Base Pointer and Other
*****
BP = 500     AX = 0    BX = 0    CX = 0

*****
      Flags
*****
flag = 0
CF = 0  PF = 0  AF = 0  ZF = 0
SF = 0  TF = 0  IF = 0  DF = 0  OF = 0

hitesh@master:~/220/lab4$
```

Set 5: Key = 30

10 20 30

```
*****
      CPU State After execution of instruction
*****

*****
      General Purpose Registers
*****
R0 = 3      R1 = 0      R2 = 2      R3 = 3
R4 = 30     R5 = 0      R6 = 0      R7 = 2
R8 = 2      R9 = 0      R10 = 0     R11 = 0
R12 = 0     R13 = 0     R14 = 0     R15 = 0

*****
      Program Counter
*****
      PC = 232

*****
      Stack Pointer
*****
      SP = 1024

*****
      Base Pointer and Other
*****
      BP = 500      AX = 0  BX = 0  CX = 0

*****
      Flags
*****
      flag = 0
      CF = 0  PF = 0  AF = 0  ZF = 0
      SF = 0  TF = 0  IF = 0  DF = 0  OF = 0

hitesh@master:~/220/lab4$
```

Set 6: Key = 60

50 60

```
*****
CPU State After execution of instruction
*****

*****
General Purpose Registers
*****
R0 = 2      R1 = 0      R2 = 0      R3 = 2
R4 = 60     R5 = 0      R6 = 0      R7 = 0
R8 = 1      R9 = 0      R10 = 0     R11 = 0
R12 = 0     R13 = 0     R14 = 0     R15 = 0

*****
Program Counter
*****
PC = 232

*****
Stack Pointer
*****
SP = 1024

*****
Base Pointer and Other
*****
BP = 500    AX = 0    BX = 0    CX = 0

*****
Flags
*****
flag = 0
CF = 0  PF = 0  AF = 0  ZF = 0
SF = 0  TF = 0  IF = 0  DF = 0  OF = 0

hitesh@master:~/220/lab4$
```


Case 3: Search on an element < middle of the array

Set 1: Key = 10

10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170

```
*****
CPU State After execution of instruction
*****

*****
General Purpose Registers
*****
R0 = 1      R1 = 0      R2 = 0      R3 = 0
R4 = 10     R5 = 0      R6 = 0      R7 = 0
R8 = 0      R9 = 0      R10 = 0     R11 = 0
R12 = 0     R13 = 0     R14 = 0     R15 = 0

*****
Program Counter
*****
PC = 232

*****
Stack Pointer
*****
SP = 1024

*****
Base Pointer and Other
*****
BP = 500     AX = 0    BX = 0    CX = 0

*****
Flags
*****
flag = 0
CF = 0  PF = 0  AF = 0  ZF = 0
SF = 0  TF = 0  IF = 0  DF = 0  OF = 0

hitesh@master:~/220/lab4$
```

Set 2: Key = 20

10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170

```
*****
CPU State After execution of instruction
*****

*****
General Purpose Registers
*****
R0 = 2      R1 = 0      R2 = 0      R3 = 2
R4 = 20     R5 = 0      R6 = 0      R7 = 0
R8 = 1      R9 = 0      R10 = 0     R11 = 0
R12 = 0     R13 = 0     R14 = 0     R15 = 0

*****
Program Counter
*****
PC = 232

*****
Stack Pointer
*****
SP = 1024

*****
Base Pointer and Other
*****
BP = 500    AX = 0    BX = 0    CX = 0

*****
Flags
*****
flag = 0
CF = 0  PF = 0  AF = 0  ZF = 0
SF = 0  TF = 0  IF = 0  DF = 0  OF = 0

hitesh@master:~/220/lab4$
```

Set 3: Key = 80

10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170

```
*****
CPU State After execution of instruction
*****

*****
General Purpose Registers
*****
R0 = 8      R1 = 0      R2 = 7      R3 = 7
R4 = 80     R5 = 0      R6 = 0      R7 = 6
R8 = 7      R9 = 0      R10 = 0     R11 = 0
R12 = 0     R13 = 0     R14 = 0     R15 = 0

*****
Program Counter
*****
PC = 232

*****
Stack Pointer
*****
SP = 1024

*****
Base Pointer and Other
*****
BP = 500     AX = 0    BX = 0    CX = 0

*****
Flags
*****
flag = 0
CF = 0  PF = 0  AF = 0  ZF = 0
SF = 0  TF = 0  IF = 0  DF = 0  OF = 0

hitesh@master:~/220/lab4$
```

Set 4: Key = 90

90 100 110 120 130 140 150 160

```
*****
CPU State After execution of instruction
*****

*****
General Purpose Registers
*****
R0 = 1      R1 = 0      R2 = 0      R3 = 0
R4 = 90     R5 = 0      R6 = 0      R7 = 0
R8 = 0      R9 = 0      R10 = 0     R11 = 0
R12 = 0     R13 = 0     R14 = 0     R15 = 0

*****
Program Counter
*****
PC = 232

*****
Stack Pointer
*****
SP = 1024

*****
Base Pointer and Other
*****
BP = 500     AX = 0    BX = 0    CX = 0

*****
Flags
*****
flag = 0
CF = 0  PF = 0  AF = 0  ZF = 0
SF = 0  TF = 0  IF = 0  DF = 0  OF = 0

hitesh@master:~/220/lab4$
```

Set 5: Key = 10

10 20 30

```
*****
CPU State After execution of instruction
*****

*****
General Purpose Registers
*****
R0 = 1      R1 = 0      R2 = 0      R3 = 0
R4 = 10     R5 = 0      R6 = 0      R7 = 0
R8 = 0      R9 = 0      R10 = 0     R11 = 0
R12 = 0     R13 = 0     R14 = 0     R15 = 0

*****
Program Counter
*****
PC = 232

*****
Stack Pointer
*****
SP = 1024

*****
Base Pointer and Other
*****
BP = 500     AX = 0    BX = 0    CX = 0

*****
Flags
*****
flag = 0
CF = 0  PF = 0  AF = 0  ZF = 0
SF = 0  TF = 0  IF = 0  DF = 0  OF = 0

hitesh@master:~/220/lab4$
```

Set 6: Key = 50

50 60

```
*****
      CPU State After execution of instruction
*****

*****
      General Purpose Registers
*****
R0 = 1      R1 = 0      R2 = 0      R3 = 0
R4 = 50     R5 = 0      R6 = 0      R7 = 0
R8 = 0      R9 = 0      R10 = 0     R11 = 0
R12 = 0     R13 = 0     R14 = 0     R15 = 0

*****
      Program Counter
*****
      PC = 232

*****
      Stack Pointer
*****
      SP = 1024

*****
      Base Pointer and Other
*****
      BP = 500      AX = 0   BX = 0   CX = 0

*****
      Flags
*****
      flag = 0
      CF = 0   PF = 0   AF = 0   ZF = 0
      SF = 0   TF = 0   IF = 0   DF = 0   OF = 0

hitesh@master:~/220/lab4$
```

Case 4: Search on an element that is not in the array

Set 1: Key = 1

10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170

```
*****
CPU State After execution of instruction
*****

*****
General Purpose Registers
*****
R0 = 0      R1 = 0      R2 = 0      R3 = 0
R4 = 1      R5 = 0      R6 = 0      R7 = 0
R8 = 0      R9 = 0      R10 = 0     R11 = 0
R12 = 0     R13 = 0     R14 = 0     R15 = 0

*****
Program Counter
*****
PC = 232

*****
Stack Pointer
*****
SP = 1024

*****
Base Pointer and Other
*****
BP = 500     AX = 0    BX = 0    CX = 0

*****
Flags
*****
flag = 0
CF = 0  PF = 0  AF = 0  ZF = 0
SF = 0  TF = 0  IF = 0  DF = 0  OF = 0

hitesh@master:~/220/lab4$
```

Set 2: Key = 180

10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170

```
*****
CPU State After execution of instruction
*****

*****
General Purpose Registers
*****
R0 = 0          R1 = 0          R2 = 17          R3 = 17
R4 = 180         R5 = 0          R6 = 0          R7 = 16
R8 = 17          R9 = 0          R10 = 0          R11 = 0
R12 = 0          R13 = 0          R14 = 0          R15 = 0

*****
Program Counter
*****
PC = 232

*****
Stack Pointer
*****
SP = 1024

*****
Base Pointer and Other
*****
BP = 500         AX = 0   BX = 0   CX = 0

*****
Flags
*****
flag = 0
CF = 0   PF = 0   AF = 0   ZF = 0
SF = 0   TF = 0   IF = 0   DF = 0   OF = 0

hitesh@master:~/220/lab4$
```


Set 3: Key = 100

10 20 30 40 50 60 70 80 90 110 120 130 140 150 160 170

```
*****
CPU State After execution of instruction
*****

*****
General Purpose Registers
*****
R0 = 0      R1 = 0      R2 = 9      R3 = 9
R4 = 100     R5 = 0      R6 = 0      R7 = 9
R8 = 9       R9 = 0      R10 = 0     R11 = 0
R12 = 0      R13 = 0     R14 = 0     R15 = 0

*****
Program Counter
*****
PC = 232

*****
Stack Pointer
*****
SP = 1024

*****
Base Pointer and Other
*****
BP = 500     AX = 0     BX = 0     CX = 0

*****
Flags
*****
flag = 0
CF = 0  PF = 0  AF = 0  ZF = 0
SF = 0  TF = 0  IF = 0  DF = 0  OF = 0

hitesh@master:~/220/lab4$
```

Set 4: Key = 70

90 100 110 120 130 140 150 160

```
*****
      CPU State After execution of instruction
*****

*****
      General Purpose Registers
*****
R0 = 0      R1 = 0      R2 = 9      R3 = 9
R4 = 100     R5 = 0      R6 = 0      R7 = 9
R8 = 9       R9 = 0      R10 = 0     R11 = 0
R12 = 0      R13 = 0     R14 = 0     R15 = 0

*****
      Program Counter
*****
PC = 232

*****
      Stack Pointer
*****
SP = 1024

*****
      Base Pointer and Other
*****
BP = 500     AX = 0     BX = 0     CX = 0

*****
      Flags
*****
flag = 0
CF = 0  PF = 0  AF = 0  ZF = 0
SF = 0  TF = 0  IF = 0  DF = 0  OF = 0

hitesh@master:~/220/lab4$
```

Set 5: Key = 2

10 20 30

```
*****
CPU State After execution of instruction
*****

*****
General Purpose Registers
*****
R0 = 0      R1 = 0      R2 = 0      R3 = 0
R4 = 2      R5 = 0      R6 = 0      R7 = 0
R8 = 0      R9 = 0      R10 = 0     R11 = 0
R12 = 0     R13 = 0     R14 = 0     R15 = 0

*****
Program Counter
*****
PC = 232

*****
Stack Pointer
*****
SP = 1024

*****
Base Pointer and Other
*****
BP = 500      AX = 0    BX = 0    CX = 0

*****
Flags
*****
flag = 0
CF = 0  PF = 0  AF = 0  ZF = 0
SF = 0  TF = 0  IF = 0  DF = 0  OF = 0

hitesh@master:~/220/lab4$
```

Set 6: Key = 100

50 60

```
*****
      CPU State After execution of instruction
*****

*****
      General Purpose Registers
*****
R0 = 0      R1 = 0      R2 = 2      R3 = 2
R4 = 100    R5 = 0      R6 = 0      R7 = 0
R8 = 2      R9 = 0      R10 = 0     R11 = 0
R12 = 0     R13 = 0     R14 = 0     R15 = 0

*****
      Program Counter
*****
PC = 232

*****
      Stack Pointer
*****
SP = 1024

*****
      Base Pointer and Other
*****
BP = 500    AX = 0    BX = 0    CX = 0

*****
      Flags
*****
flag = 0
CF = 0  PF = 0  AF = 0  ZF = 0
SF = 0  TF = 0  IF = 0  DF = 0  OF = 0

hitesh@master:~/220/lab4$
```