Computer Science and Engineering
IIIT Kalyani, West Bengal

Compilers Design Laboratory (CS 511)
(Autumn: 2019 - 2020)

3rd Year CSE: 5th Semester

Assignment - 1 *Marks: 10*

Assignment Out: *19th July, 2019* Report on or before: *26th July, 2019*

1. A program is split in two files - the function `int main()`, written in C, is in `main1.c` and the function `int what(int data[], int no)`, written in the assembly language of x86-64, is in `what.s`. The files can be compiled as `$ cc -Wall main1.c what.s` to get the executable file `a.out`.
   Compile the code and run it to see what it does and convince yourself that it works.

2. Write comments with every line of the assembly language code of `what.s` explaining its action.

3. Reconstruct a C function `int what(int data[], int no)`, equivalent to the code of `what.s`. Assume the following information for the reconstruction:

   - The first argument of the function is passed through the register `rdi`.
   - The second argument is passed through the register `esi` (lower 32-bits of `rsi`).
   - The names associated to memory locations in the function `int what(int data[], int no)` are follows:
     ```
     Mem[rbp-24]:  int data[]
     Mem[rbp-28]:  int no
     Mem[rbp-12]:  int i
     Mem[rbp-8]:  int j
     Mem[rbp-4]:  int temp
     ```

4. Let the file name of the C code of `int what(int data[], int no)` be `what1.c`. You can recompile it with `main1.c` as
   `$ cc -Wall main1.c what1.c`
   and test it.

5. Write your name, roll number, registration number, and any other comments at the top of both `what.s` and `what1.c` files.
   `# Name ...` etc. for `what.s` file, and
   `// Name ...` etc. for `what1.c` file.

6. Prepare a file with name `<Roll No>.1.tar` with `what.s` and `what1.c` with the command,
   `$ tar cvf <Roll No>.1.tar what.s what1.c`
   Do not use any other name for any file.

7. Reading material is available at `http://cse.iitkgp.ac.in/~goutam/`

```
// main1.c
#include <stdio.h>
#define MAXNO 100
void what(int [], int);
int main()  // main1.c
{
    int no = 0, i ;
    int data[MAXNO] ;

    printf("Enter data, terminate with Ctrl+D\n") ;
    while(scanf("%d", &data[no]) != EOF) ++no;
    what(data, no) ;
    printf("Data in sorted order is: ") ;
    for(i = 0; i < no; ++i) printf("%d ", data[i]);
    putchar(’\n’) ;
    return 0 ;
} // main1.c
```

```
# what.s
#
  .file  "what.c"
  .text
  .globl  what
  .type  what, @function
what:
.LFB0:
  pushq  %rbp
  movq  %rsp, %rbp
  movq  %rdi, -24(%rbp)
  movl  %esi, -28(%rbp)
  movl  $1, -12(%rbp)
  jmp  .L2
.L8:
  movl  -12(%rbp), %eax
  cltq
  leaq  0(,%rax,4), %rdx
  movq  -24(%rbp), %rax
  addq  %rdx, %rax
  movl  (%rax), %eax
  movl  %eax, -4(%rbp)
  movl  -12(%rbp), %eax
  subl  $1, %eax
  movl  %eax, -8(%rbp)
  jmp  .L3
.L7:
  movl  -8(%rbp), %eax
  cltq
  leaq  0(,%rax,4), %rdx
  movq  -24(%rbp), %rax
  addq  %rdx, %rax
  movl  (%rax), %eax
  cmpl  %eax, -4(%rbp)
  jle  .L9
  movl  -8(%rbp), %eax
  cltq
  leaq  0(,%rax,4), %rdx
  movq  -24(%rbp), %rax
  addq  %rdx, %rax
  movl  -8(%rbp), %edx
  movslq  %edx, %rdx
  addq  $1, %rdx
  leaq  0(,%rdx,4), %rcx
  movq  -24(%rbp), %rdx
  addq  %rcx, %rdx
  movl  (%rax), %eax
  movl  %eax, (%rdx)
  subl  $1, -8(%rbp)
.L3:
  cmpl  $0, -8(%rbp)
  jns  .L7
  jmp  .L6
.L9:
  nop
.L6:
  movl  -8(%rbp), %eax
  cltq
  addq  $1, %rax
  leaq  0(,%rax,4), %rdx
  movq  -24(%rbp), %rax
  addq  %rax, %rdx
  movl  -4(%rbp), %eax
  movl  %eax, (%rdx)
  addl  $1, -12(%rbp)
.L2:
  movl  -12(%rbp), %eax
  cmpl  -28(%rbp), %eax
  jl  .L8
  nop
  popq  %rbp
  ret
.LFE0:
  .size  what, .-what
  .ident  "GCC: (Ubuntu 7.4.0-1ubuntu1~18.04.1) 7.4.0"
  .section  .note.GNU-stack,"",@progbits
```