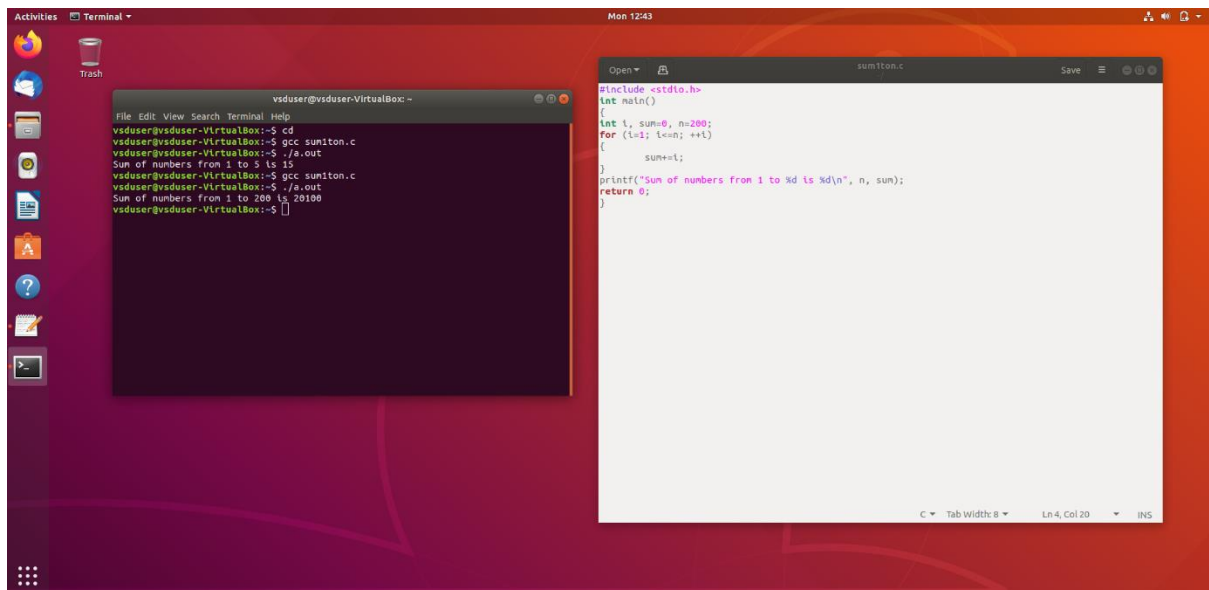
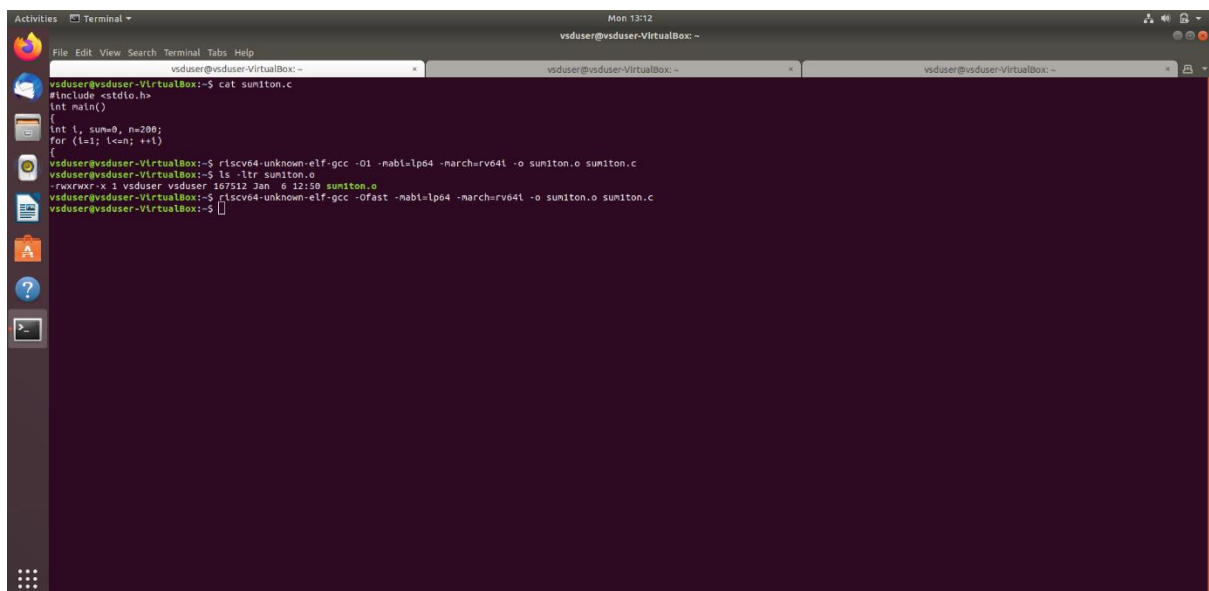


TASK 1



```
vsduser@vsduser-VirtualBox: ~  
vsduser@vsduser-VirtualBox:~$ cd  
vsduser@vsduser-VirtualBox:~$ gcc sum1ton.c  
vsduser@vsduser-VirtualBox:~$ ./a.out  
Sum of numbers from 1 to 5 is 15  
vsduser@vsduser-VirtualBox:~$ gcc sum1ton.c  
vsduser@vsduser-VirtualBox:~$ ./a.out  
Sum of numbers from 1 to 200 is 20100  
vsduser@vsduser-VirtualBox:~$  
  
sum1ton.c  
#include <stdio.h>  
int main()  
{  
    int i, sum=0, n=200;  
    for (i=1; i<=n; ++i)  
    {  
        sum+=i;  
    }  
    printf("Sum of numbers from 1 to %d is %d\n", n, sum);  
    return 0;  
}
```

For the first task we wrote a C program to calculate sum of 'n' numbers and obtained the output. This was to compare the compilation of the program in gcc vs riscv.



```
vsduser@vsduser-VirtualBox: ~  
vsduser@vsduser-VirtualBox:~$ cat sum1ton.c  
#include <stdio.h>  
{  
    int i, sum=0, n=200;  
    for (i=1; i<=n; ++i)  
    {  
        sum+=i;  
    }  
    printf("Sum of numbers from 1 to %d is %d\n", n, sum);  
    return 0;  
}  
vsduser@vsduser-VirtualBox:~$ riscv64-unknown-elf-gcc -O1 -mabi=lp64 -march=rv64i -o sum1ton.o sum1ton.c  
vsduser@vsduser-VirtualBox:~$ ls -ltr sum1ton.o  
-rwxrwxr-x 1 vsduser vsduser 107512 Jan 6 12:50 sum1ton.o  
vsduser@vsduser-VirtualBox:~$ riscv64-unknown-elf-gcc -Ofast -mabi=lp64 -march=rv64i -o sum1ton.o sum1ton.c  
vsduser@vsduser-VirtualBox:~$
```

Here we ran the same program on the risc-v processor and observed how the register is allocated for the data and how many bits of the register is occupied by the main program

```

10170: 00012517      autpc    a0,0x12
10174: e9050513      addi     a0,a0,-368 # 22000 <__FRAME_END__>
10178: 00000317      autpc    t1,0x0
1017c: 00000067      jr       zero # 0 <register_fini-0x100b0>
10180: 00000067      ret

0000000000010184 <.main>:
10184: ff010113      addi     sp,sp,-16
10188: 00113423      sd       ra,0(sp)
1018c: 0c800793      li       a5,200
10190: fff7879b      addiw    a5,a5,-1
10194: fe079ee3      bnez     a5,10190 <.main-0xc>
10198: 00005637      lut      a2,0x5
1019c: e9460613      addi     a2,a2,-380 # 4e84 <register_fini-0xb22c>
101a0: 0c800593      li       a1,200
101a4: 00021537      lut      a0,0x21
101a8: 19050513      addi     a0,a0,400 # 21190 <__clzdi2+0x48>
101ac: 26c000ef      jal      ra,10418 <printf>
101b0: 00000513      li       a0,0
101b4: 00013083      ld       ra,0(sp)
101b8: 01010113      addi     sp,sp,16
101bc: 00000067      ret

00000000000101c0 <.atexit>:
101c0: 00050593      mv       a1,a0
101c4: 00000693      ll       a3,0
101c8: 00000613      lw       a2,0
101cc: 00000513      li       a0,0
101d0: 4390206f      j        12e08 <__register_exitproc>

00000000000101d4 <.exit>:
101d4: ff010113      addi     sp,sp,-16
101d8: 00000593      li       a1,0
101dc: 00013023      sd       s0,0(sp)
101e0: 00113423      sd       ra,0(sp)
101e4: 00050413      mv       s0,a0
101e8: 4c0000ef      jal      ra,12eb4 <__call_exitprocs>
101ec: 74b18793      addi     a5,gp,1864 # 23150 <_global_inpure_ptr>
101f0: 0007b503      ld       a0,0(a5)
101f4: 05053793      ld       a5,0(a0)
101f8: 00078463      beqz     a5,10200 <.exit+0x2c>
101fc: 000780e7      jalr     a5
10200: 00040513      mv       a0,a0
10204: 7410e0ef      jal      ra,1f44 <.exit>

```

The above picture shows the register allocation of the main program when port O1 was considered. We can see a difference of 4bits between each register that is each register can store upto 4bits of data.

```

sun1ton.o: file format elf64-littlelscv

Disassembly of section .text:

00000000000100b0 <.main>:
100b0: 00005637      lut      a2,0x5
100b4: 00021537      lut      a0,0x21
100b8: ff010113      addi     sp,sp,-16
100bc: e9460613      addi     a2,a2,-380 # 4e84 <.main-0xb22c>
100c0: 0c800593      li       a1,200
100c4: 19050513      addi     a0,a0,384 # 21180 <__clzdi2+0x44>
100c8: 00113423      sd       ra,0(sp)
100cc: 340000ef      jal      ra,1040c <printf>
100d0: 00013083      ld       ra,0(sp)
100d4: 00000513      li       a0,0
100d8: 01010113      addi     sp,sp,16
100dc: 00000067      ret

00000000000100e0 <.register_fini>:
100e0: ffff0797      autpc    a5,0xffff0
100e4: f2078793      addi     a5,a5,-224 # 0 <.main-0x100b0>
100e8: 00078063      beqz     a5,100f8 <.register_fini+0x18>
100ec: 00000517      autpc    a0,0x0
100f0: 11050513      addi     a0,a0,272 # 101fc <__libc_fini_array>
100f4: 0c00006f      j        101b4 <.atexit>
100f8: 00000067      ret

00000000000100fc <.start>:
100fc: 00013197      autpc    gp,0x13
10100: 90c18193      addi     gp,gp,-1780 # 22a08 <_global_pointers>
10104: 77018513      addi     a0,gp,1904 # 23178 <_edata>
10108: 00013617      autpc    a2,0x13
1010c: 10060613      addi     a2,a2,256 # 23208 <__BSS_END__>
10110: 40a06633      sub      a2,a2,a0
10114: 00000593      li       a1,0
10118: 104000ef      jal      ra,102ec <memset>
1011c: 00000517      autpc    a0,0x0
10120: 00050513      addi     a0,a0,224 # 101fc <__libc_fini_array>
10124: 0000006f      jal      ra,101b4 <.atexit>
10128: 130000ef      jal      ra,10258 <__libc_lnt_array>
1012c: 00012003      lw       a0,0(sp)
10130: 00018593      addi     a1,sp,8
10134: 00000613      li       a2,0

```

The above picture shows the register allocation of the main program when port 'Ofast' was considered.