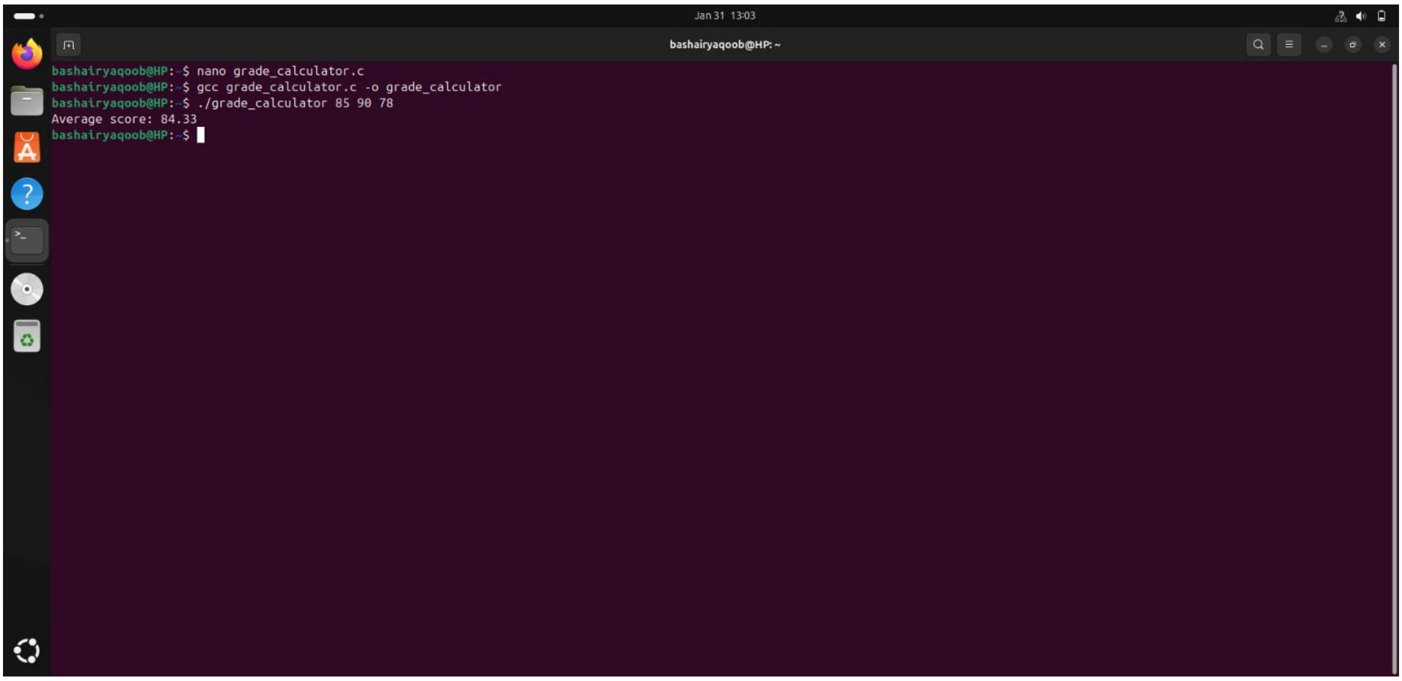
OS LAB 2:

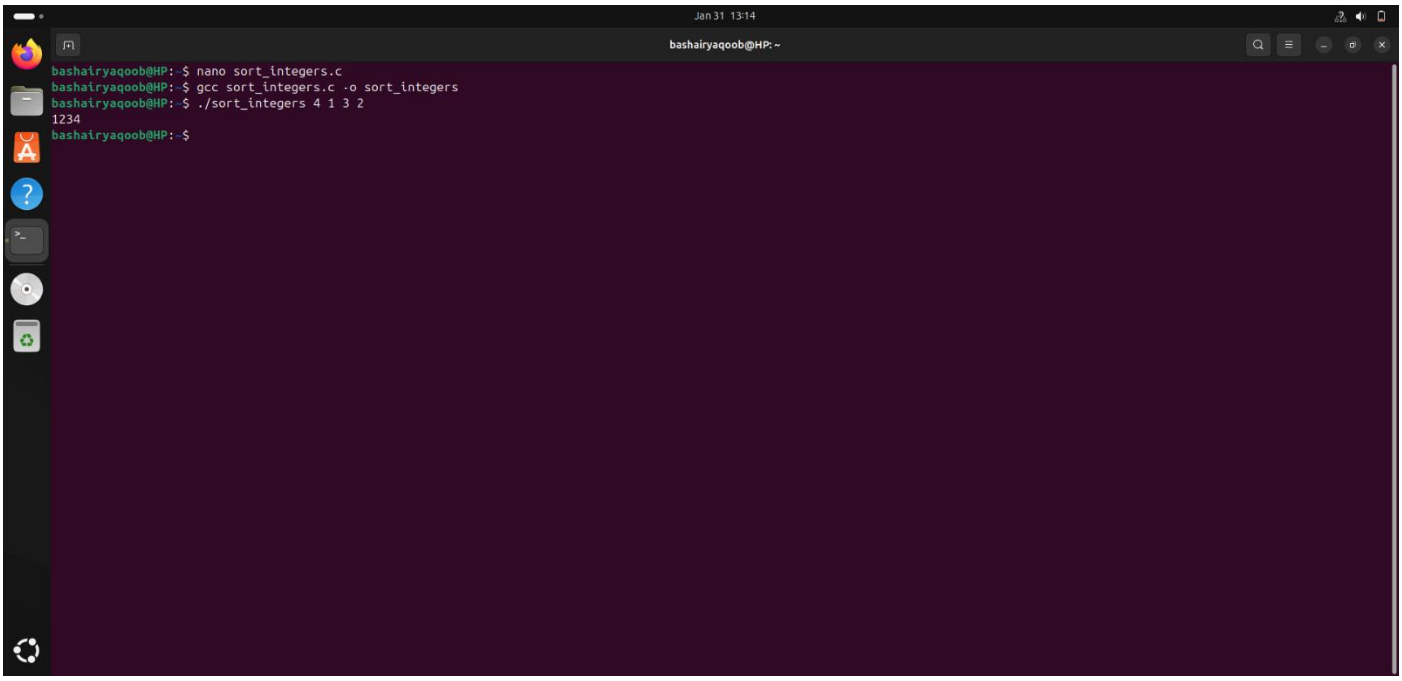
24k-0810

Q1: Write a C program that accepts student scores (out of 100) as command line parameters and calculates their average grade. Implement robust error checking for missing or incorrectly formatted parameters. For example, if the user runs the program with ./grade_calculator 85 90 78, it should compute the average of these scores and display it.



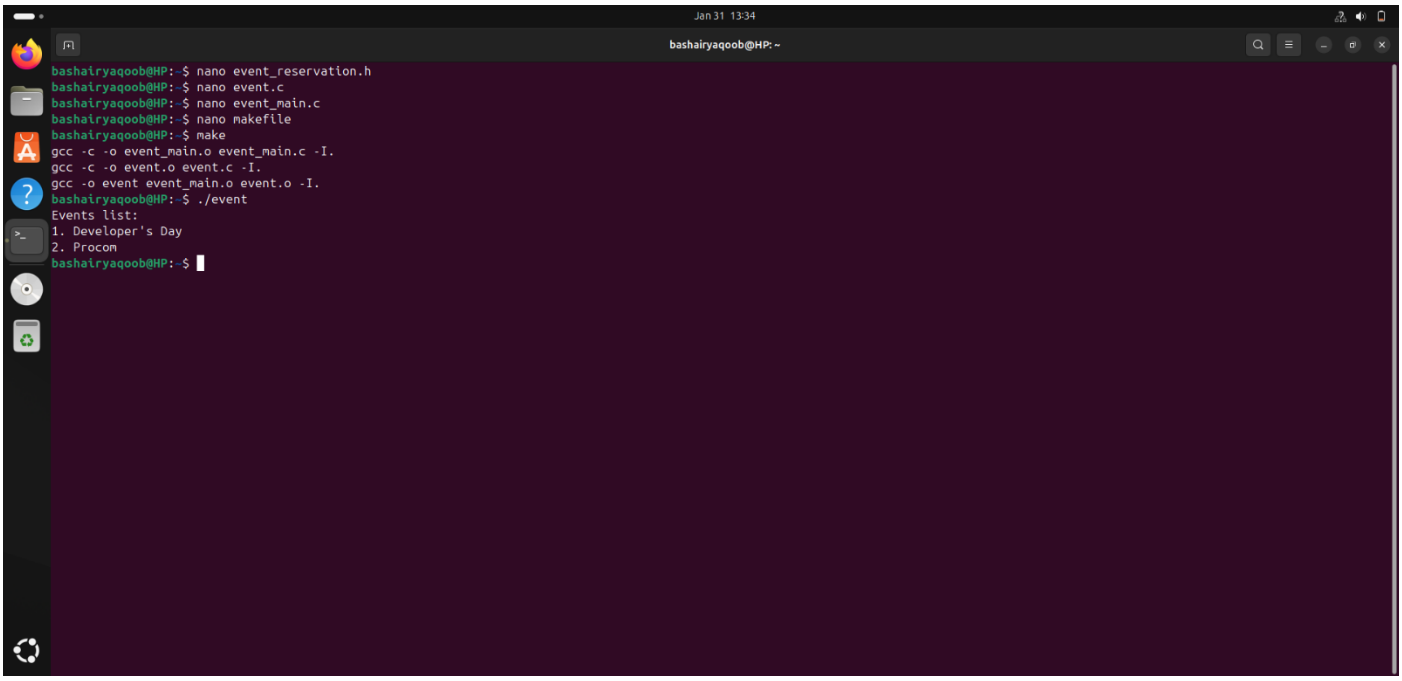
```
bashairyaqoob@HP:~$ nano grade_calculator.c
bashairyaqoob@HP:~$ gcc grade_calculator.c -o grade_calculator
bashairyaqoob@HP:~$ ./grade_calculator 85 90 78
Average score: 84.33
bashairyaqoob@HP:~$ 
```

Q2: Write a simple C program that accepts a series of integers as command line parameters, stores them in an array, sorts the array in ascending order, and prints the sorted array to the screen. For example, if the input is ./sort_integers 4 1 3 2, the output should be 1 2 3 4.

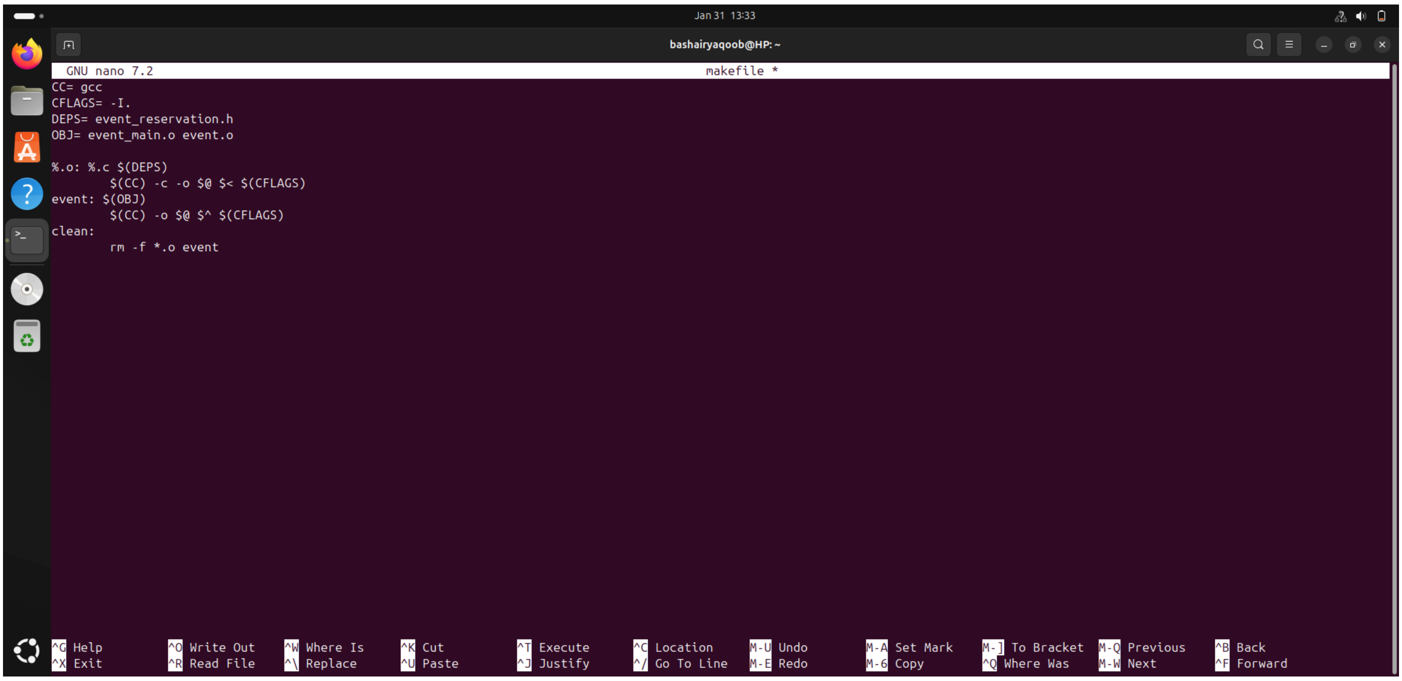


```
bashairyaqoob@HP:~$ nano sort_integers.c
bashairyaqoob@HP:~$ gcc sort_integers.c -o sort_integers
bashairyaqoob@HP:~$ ./sort_integers 4 1 3 2
1234
bashairyaqoob@HP:~$
```

Q3: Event Reservation System (Basic) Create a header file named event_reservation.h that declares functions for an Event Reservation System. Declare functions for adding and displaying events. Implement these functions in a source file named event.c. Provide a simple main function in another source file named event_main.c to showcase adding and displaying events. Now compile all classes using makefile.

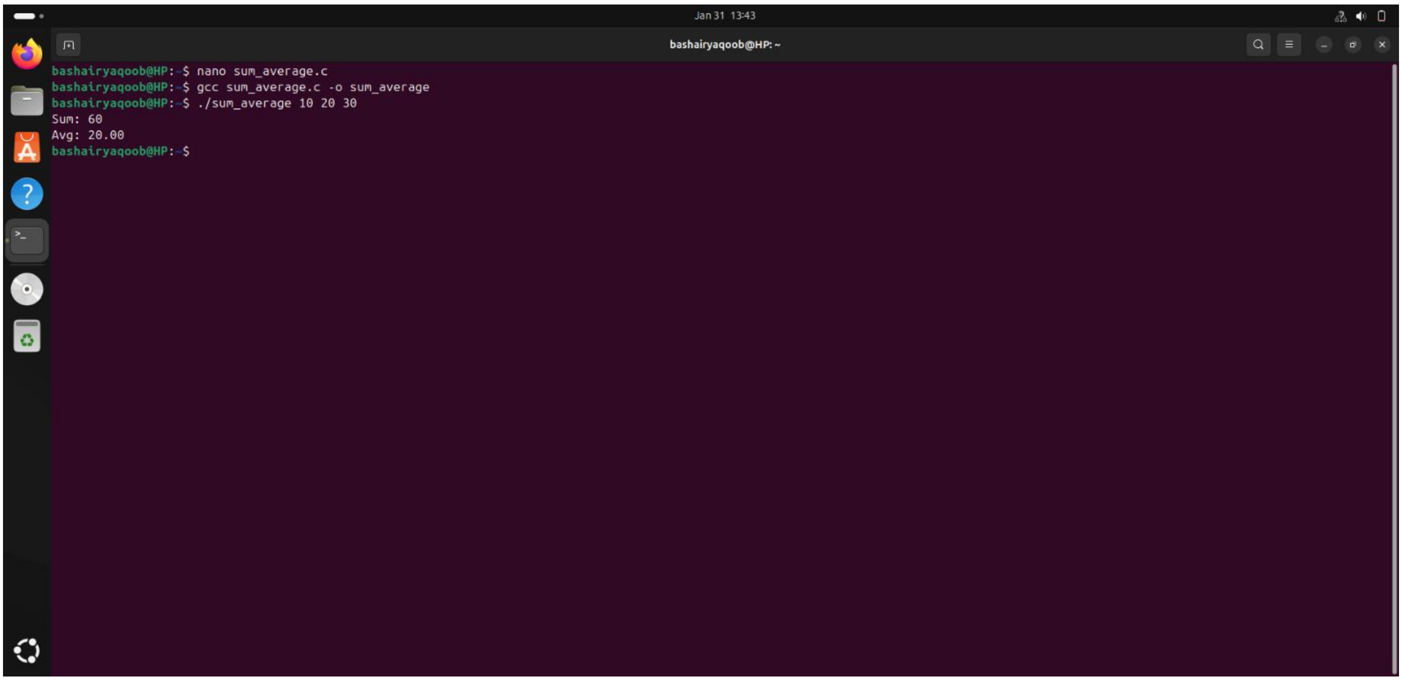


```
bashairyaqoob@HP:~$ nano event_reservation.h
bashairyaqoob@HP:~$ nano event.c
bashairyaqoob@HP:~$ nano event_main.c
bashairyaqoob@HP:~$ nano makefile
bashairyaqoob@HP:~$ make
gcc -c -o event_main.o event_main.c -I.
gcc -c -o event.o event.c -I.
gcc -o event event_main.o event.o -I.
bashairyaqoob@HP:~$ ./event
Events list:
1. Developer's Day
2. Procom
bashairyaqoob@HP:~$ 
```



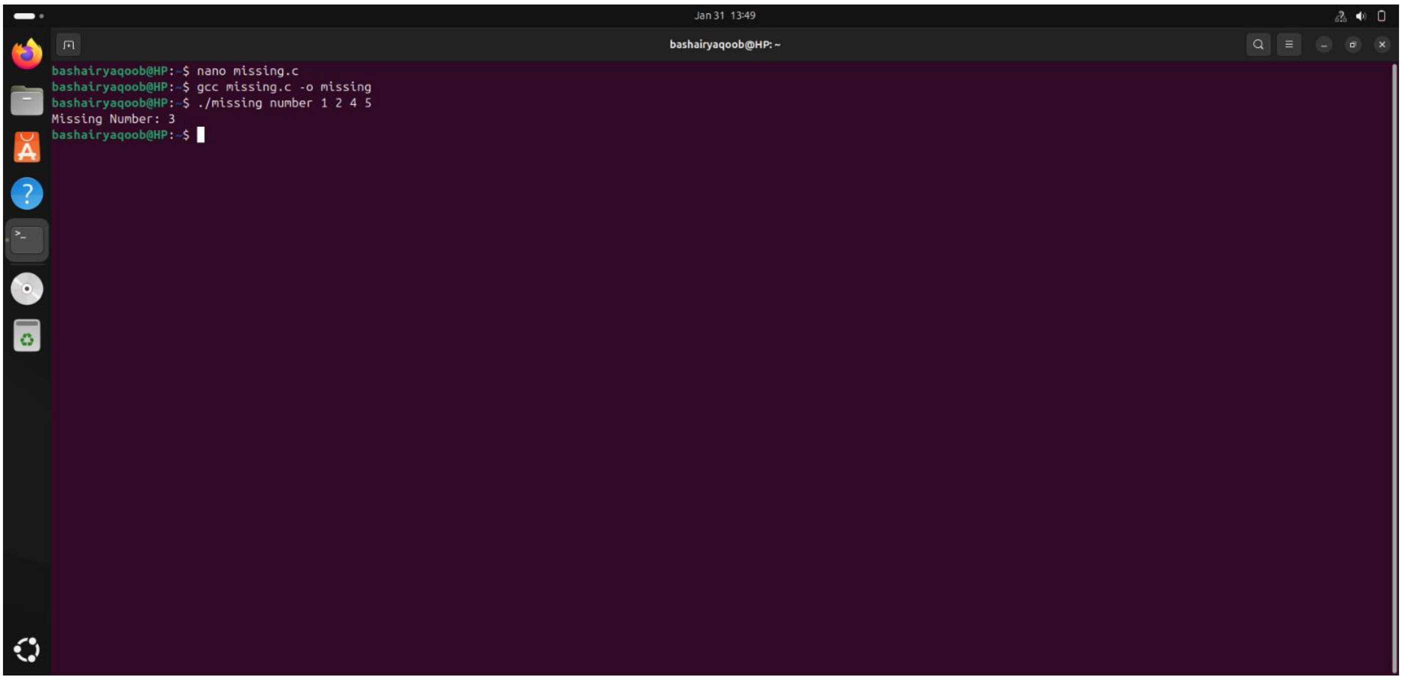
```
CC= gcc
CFLAGS= -I.
DEPS= event_reservation.h
OBJ= event_main.o event.o

%.o: %.c $(DEPS)
        $(CC) -c -o $@ $< $(CFLAGS)
event: $(OBJ)
        $(CC) -o $@ $^ $(CFLAGS)
clean:
        rm -f *.o event
```

Q4: Write a C program that accepts integers as command line parameters, stores them in an array, and prints the sum and average of those integers. Implement error checking for missing parameters and non-integer inputs. For example, running ./sum_average 10 20 30 should output the sum (60) and average (20).



```
bashairyaqoob@HP:~$ nano sum_average.c
bashairyaqoob@HP:~$ gcc sum_average.c -o sum_average
bashairyaqoob@HP:~$ ./sum_average 10 20 30
Sum: 60
Avg: 20.00
bashairyaqoob@HP:~$
```

Q5: Write a C program that accepts a series of integers as command line parameters, stores them in an array, computes the missing element in a sequence (assuming the sequence is a continuous range of numbers), and prints the missing element. For example, for input 1 2 4 5, the program should output 3.



```
bashairyaqoob@HP:~$ nano missing.c
bashairyaqoob@HP:~$ gcc missing.c -o missing
bashairyaqoob@HP:~$ ./missing number 1 2 4 5
Missing Number: 3
bashairyaqoob@HP:~$
```