

Assignment-1 assumptions document

2021101113

CS2.201: Computer Systems Organization

arithmetic, conditional, looping components,
procedure calls, and conditional jumps in x86-64

Gowlapalli Rohit

All these commands are tested on Ubuntu Version 20.04.3 LTS (Focal Fossa)

```
2021101113
├── q1
│   ├── q1.c
│   └── q1.s
├── q2
│   ├── q2.c
│   └── q2.s
├── q3
│   ├── q3.c
│   └── q3.s
├── q4
│   ├── q4.c
│   └── q4.s
└── q5
    ├── q5.c
    └── q5.s
```

- q1

```
gcc q1.c q1.s
```

It is assumed that when $N=0$, "-1 -1" is printed on the screen

It is assumed that remainder is always non-negative (if $M=-20$ and $N=-6$ the
(quotient, remainder) = (4, 4))

inbuilt `idivq` and other similar operations are not used for this question

It is assumed all the integer variables to be long long int

In case of error or invalid input '-1' is returned by both fun1,fun2

Overflow cases are handled

- q2

`gcc q2.c q2.s`

It is assumed that $\gcd(|a|,0)=\gcd(0,|a|)=|a|$

It is assumed that $\gcd(a,b)=\gcd(|a|,|b|)$

It is assumed all the integer variables to be long long int

In case of error or invalid input '-1' is returned by fun3

Overflow cases are handled

- q3

`gcc q3.c q3.s`

If N is prime ,1 is returned by fun4 else -1 is returned,printf statements are used to print necessary outputs accordingly

It is assumed that if $N \leq 1$,"FALSE" is printed on screen

It is assumed all the integer variables to be long long int

In case of error or invalid input '-1' is returned by fun4

Overflow cases are handled

- q4

`gcc q4.c q4.s`

It is assumed that if $N \leq 1$, -1 is returned by fun4,fun5

fun4 from q3 is reused to check if a number is prime

It is assumed all the integer variables to be long long int

In case of error or invalid input '-1' is returned by both fun4,fun5

Overflow cases are handled

- q5

`gcc q5.c q5.s`

It is assumed that if $N < 1$, -1 is returned by fun6

It is assumed all the integer variables to be long long int

In case of error or invalid input '-1' is returned by fun6

Overflow cases are handled

If Overflow occurs, then actual_result is replaced with $(\text{actual_result}) \bmod (\text{LONG_MAX} + 1)$

Specifically, the square of integers from 1 to n are computed and then the values are summed up

`LONG_MAX=9223372036854775807`