[1.1] Give a general description of what you happens with the following " log/show-ref " commands. What do they all have in common? Record what the first 6 characters are of the tip commit sha.

'git log' shows the commit logs. It shows the commits made in the repository in the reverse order i.e. the most recent commits are shown first.

'git show-ref' displays references available in a local repository along with the associated commit IDs.

First 6 characters of the tip commit sha: ae3760

[1.2] Give a general description of what you happens to the repository when you used the soft and hard reset commands below?

' --soft' parameter tells Git to reset HEAD to another commit. If we specify --soft Git will stop there and nothing else will change.
' --hard' parameter will blow out everything – it resets HEAD back to another commit, resets the index to match it, and resets the working copy to match it as well. This is the more dangerous of the commands and is where you can cause damage. Data might get lost here.

[1.3] Give a general description of what you see with the "diff/status/log" commands below. What do they all have in common?

diff - Show changes between commits, commit and working tree, etc

status - This command displays the state of the working directory and the staging area. It lets you see which changes have been staged, which haven't, and which files aren't being tracked by Git.

log - Shows the commit logs. It shows the commits made in the repository in the reverse order i.e. the most recent commits are shown first.

diff/status/log all the three commands show commit logs.

[1.4] Use the previous "diff status, show-ref and log" commands to confirm that you are at the tip. Does the tip commit sha you are observing on your branch match the one from part 1?

No

[1.5] Given the list of commands below, do any of these differ in output? What do all of these commands have in common? Use git show-ref <reference> to help.

$ git diff HEAD~2 HEAD
$ git diff prob_1~2 prob_1
$ git diff prob_1~2 origin/develop
$ git diff origin/develop~2 HEAD

All of these commands show the commit logs.

[2.2] Run through the following sequence of making a commit and annotate what you see with your git status commands. What do the different git status command outputs show after each step? Why did they differ?

1.When we run Git status for the first time it shows no untracked changes and local is up to date.
2. Directory is changed to git-assignment/students_sum2017/hw-simple-change
3. We make changes to favorites.json
4.Nothing added to commit but untracked files present.
5. Add favorites.json
6.Shows untracked files present but the local is up to date with the origin and changes to be committed are shown.

7. Use git commit
8.Shows the commit message and no files to commit. Then shows commit is successful.

Why did they differ? Commit status changes due to adding favorites.json

[2.3] Try to push your commit (which is based off of an older commit to develop). What happens when you try to push your commit with the following command? Explain.
$ git push origin prob_2:develop

The operation fails. Updates were rejected because the remote contains work that we do not have locally. This is usually caused by another repository pushing to the same ref.

[2.4] Attempt to do a rebase by running the following commands. What changes to the commands make to your history? Look at your parent commits. Did you have a conflict? How can you resolve this without destroying other's work?

Yes, I got a conflict. We can resolve this without destroying other's work by doing the following steps:

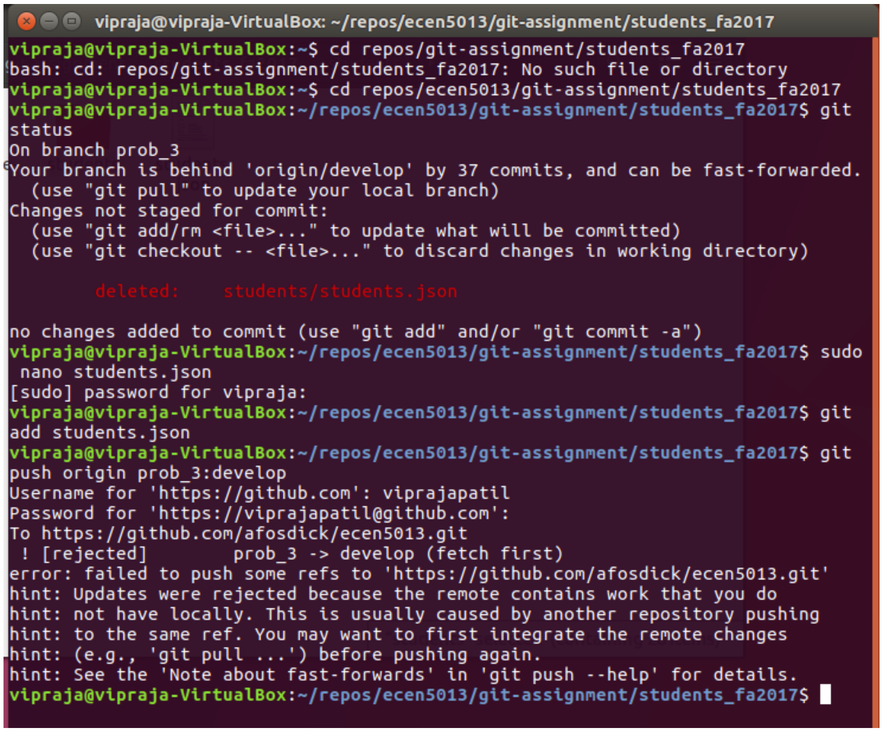
1. $ git fetch origin
2. $ git rebase --onto origin/develop prob_2~ prob_2
3. edit favorites.json and add it using git add favorites.json
4. $ git commit -m “Adding Favorites, Rebase Completed”
5. $ git push origin prob_2:develop

[2.5] How does git fetch differ from git pull? Why might you prefer to use git fetch over git pull?

'git fetch' only downloads new data from a remote repository - but it doesn't integrate any of this new data into your working files. 'git pull' in contrast, is used with a different goal in mind: to update your current HEAD branch with the latest changes from the remote server. This means that pull not only downloads new data; it also directly it into your current working copy files.
I would prefer git fetch because git fetch only downloads latest changes where as git pull also merges them.

[3.1] Did this Fail? What was the error? Why? How can you fix this (provide the commands you need in order to do this)? Fix your change so that it gets pushed to tip using a rebase command. Grab a screenshot of your rebase command succeeding in addition to a copy of the oneline log to show the time difference. Push your change.

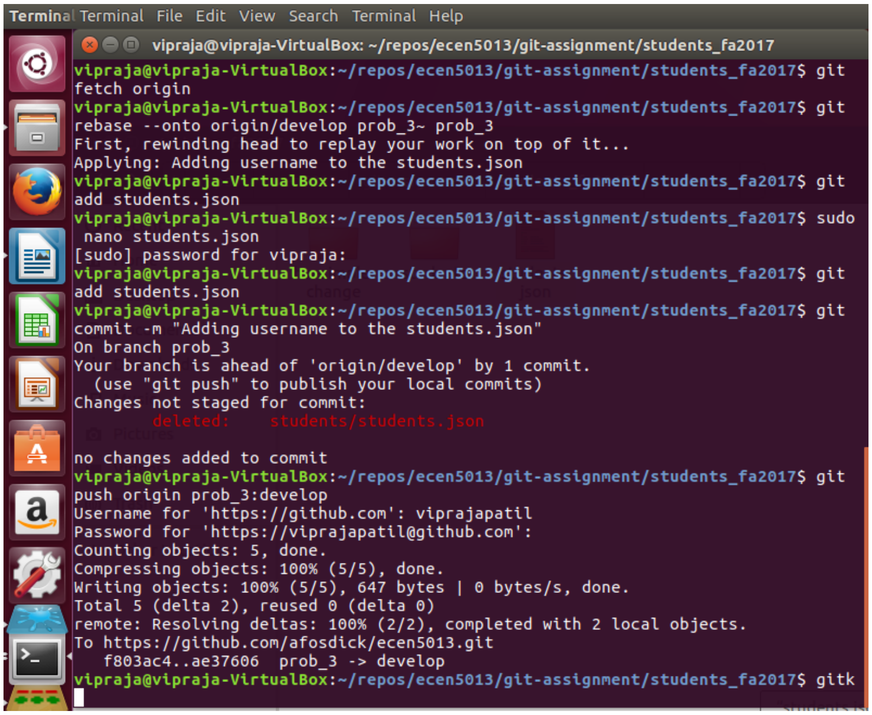
Error



vipraja@vipraja-VirtualBox: ~/repos/ecen5013/git-assignment/students_fa2017

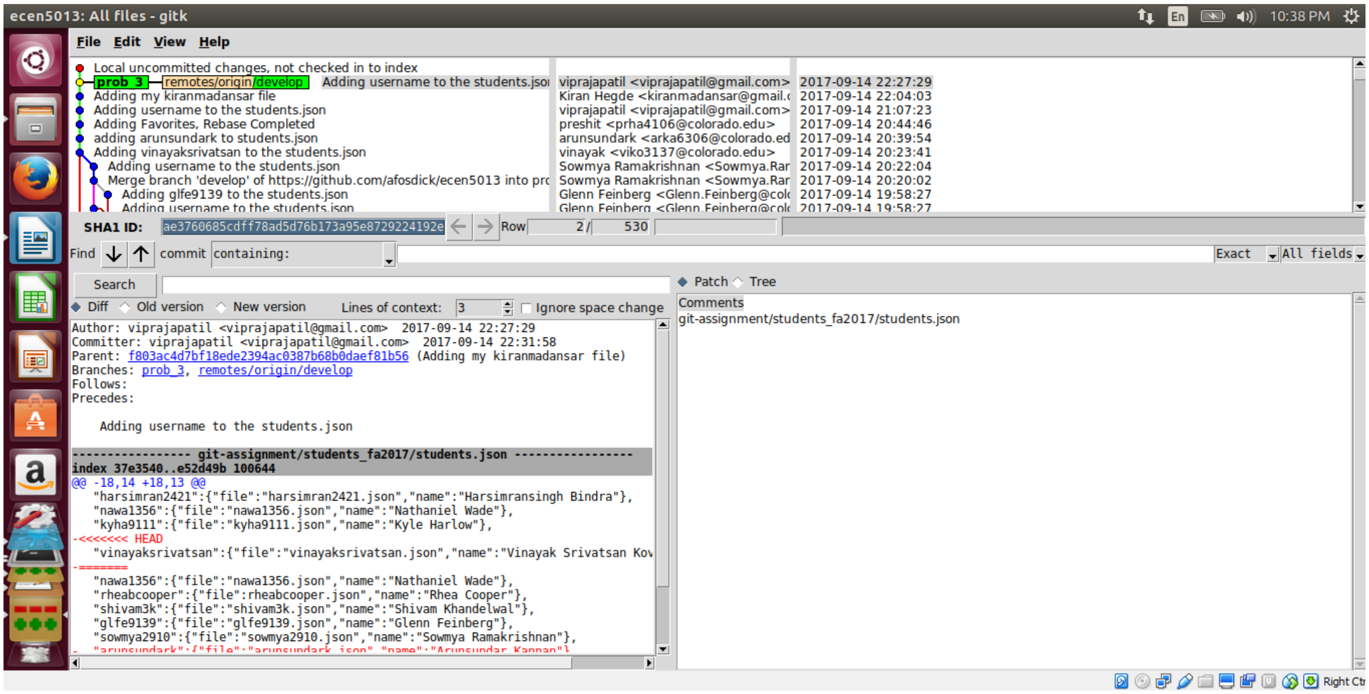
```
vipraja@vipraja-VirtualBox:~$ cd repos/git-assignment/students_fa2017
bash: cd: repos/git-assignment/students_fa2017: No such file or directory
vipraja@vipraja-VirtualBox:~$ cd repos/ecen5013/git-assignment/students_fa2017
vipraja@vipraja-VirtualBox:~/repos/ecen5013/git-assignment/students_fa2017$ git
status
On branch prob_3
Your branch is behind 'origin/develop' by 37 commits, and can be fast-forwarded.
  (use "git pull" to update your local branch)
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        deleted:    students/students.json

no changes added to commit (use "git add" and/or "git commit -a")
vipraja@vipraja-VirtualBox:~/repos/ecen5013/git-assignment/students_fa2017$ sudo
 nano students.json
[sudo] password for vipraja:
vipraja@vipraja-VirtualBox:~/repos/ecen5013/git-assignment/students_fa2017$ git
add students.json
vipraja@vipraja-VirtualBox:~/repos/ecen5013/git-assignment/students_fa2017$ git
push origin prob_3:develop
Username for 'https://github.com': viprajapatil
Password for 'https://viprajapatil@github.com':
To https://github.com/afosdick/ecen5013.git
 ! [rejected]        prob_3 -> develop (fetch first)
error: failed to push some refs to 'https://github.com/afosdick/ecen5013.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
vipraja@vipraja-VirtualBox:~/repos/ecen5013/git-assignment/students_fa2017$
```

solution



Terminal Terminal File Edit View Search Terminal Help
vipraja@vipraja-VirtualBox: ~/repos/ecen5013/git-assignment/students_fa2017

```
vipraja@vipraja-VirtualBox:~/repos/ecen5013/git-assignment/students_fa2017$ git
fetch origin
vipraja@vipraja-VirtualBox:~/repos/ecen5013/git-assignment/students_fa2017$ git
rebase --onto origin/develop prob_3~ prob_3
First, rewinding head to replay your work on top of it...
Applying: Adding username to the students.json
vipraja@vipraja-VirtualBox:~/repos/ecen5013/git-assignment/students_fa2017$ git
add students.json
vipraja@vipraja-VirtualBox:~/repos/ecen5013/git-assignment/students_fa2017$ sudo
 nano students.json
[sudo] password for vipraja:
vipraja@vipraja-VirtualBox:~/repos/ecen5013/git-assignment/students_fa2017$ git
add students.json
vipraja@vipraja-VirtualBox:~/repos/ecen5013/git-assignment/students_fa2017$ git
commit -m "Adding username to the students.json"
On branch prob_3
Your branch is ahead of 'origin/develop' by 1 commit.
  (use "git push" to publish your local commits)
Changes not staged for commit:
        deleted:    students/students.json

no changes added to commit
vipraja@vipraja-VirtualBox:~/repos/ecen5013/git-assignment/students_fa2017$ git
push origin prob_3:develop
Username for 'https://github.com': viprajapatil
Password for 'https://viprajapatil@github.com':
Counting objects: 5, done.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 647 bytes | 0 bytes/s, done.
Total 5 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/afosdick/ecen5013.git
   f803ac4..ae37606  prob_3 -> develop
vipraja@vipraja-VirtualBox:~/repos/ecen5013/git-assignment/students_fa2017$ gitk
```



```
Local uncommitted changes, not checked in to index
prob 3  remotes/origin/develop  Adding username to the students.json   viprajapatil <viprajapatil@gmail.com>   2017-09-14 22:27:29
Adding my kiranmadansar file                                            Kiran Hegde <kiranmadansar@gmail.c      2017-09-14 22:04:03
Adding username to the students.json                                    viprajapatil <viprajapatil@gmail.com>   2017-09-14 21:07:23
Adding Favorites, Rebase Completed                                      preshit <prha4106@colorado.edu>         2017-09-14 20:44:46
adding arunsundark to students.json                                     arunsundark <arka6306@colorado.ed       2017-09-14 20:39:54
Adding vinayaksrivatsan to the students.json                            vinayak <viko3137@colorado.edu>         2017-09-14 20:23:41
  Adding username to the students.json                                  Sowmya Ramakrishnan <Sowmya.Rar         2017-09-14 20:22:04
  Merge branch 'develop' of https://github.com/afosdick/ecen5013 into prc Sowmya Ramakrishnan <Sowmya.Rar      2017-09-14 20:20:02
    Adding glfe9139 to the students.json                                Glenn Feinberg <Glenn.Feinberg@colc     2017-09-14 19:58:27
    Adding username to the students.json                                Glenn Feinberg <Glenn.Feinberg@colc     2017-09-14 19:58:27
```

```
SHA1 ID: ae3760685cdff78ad5d76b173a95e8729224192e   Row 2 / 530
Find  commit containing:                                  Exact  All fields
◆ Diff  ◇ Old version  ◇ New version   Lines of context: 3   □ Ignore space change
```

```
Author: viprajapatil <viprajapatil@gmail.com>  2017-09-14 22:27:29
Committer: viprajapatil <viprajapatil@gmail.com>  2017-09-14 22:31:58
Parent: f803ac4d7bf18ede2394ac0387b68b0daef81b56 (Adding my kiranmadansar file)
Branches: prob_3, remotes/origin/develop
Follows:
Precedes:

    Adding username to the students.json

---------------- git-assignment/students_fa2017/students.json ----------------
index 37e3540..e52d49b 100644
@@ -18,14 +18,13 @@
   "harsimran2421":{"file":"harsimran2421.json","name":"Harsimransingh Bindra"},
   "nawa1356":{"file":"nawa1356.json","name":"Nathaniel Wade"},
   "kyha9111":{"file":"kyha9111.json","name":"Kyle Harlow"},
-<<<<<<< HEAD
   "vinayaksrivatsan":{"file":"vinayaksrivatsan.json","name":"Vinayak Srivatsan Kov
-=======
   "nawa1356":{"file":"nawa1356.json","name":"Nathaniel Wade"},
   "rheabcooper":{"file":rheabcooper.json","name":"Rhea Cooper"},
   "shivam3k":{"file":"shivam3k.json","name":"Shivam Khandelwal"},
   "glfe9139":{"file":"glfe9139.json","name":"Glenn Feinberg"},
   "sowmya2910":{"file":"sowmya2910.json","name":"Sowmya Ramakrishnan"},
-  "arunsundark":{"file":"arunsundark.json","name":"Arunsundar Kannan"},
```

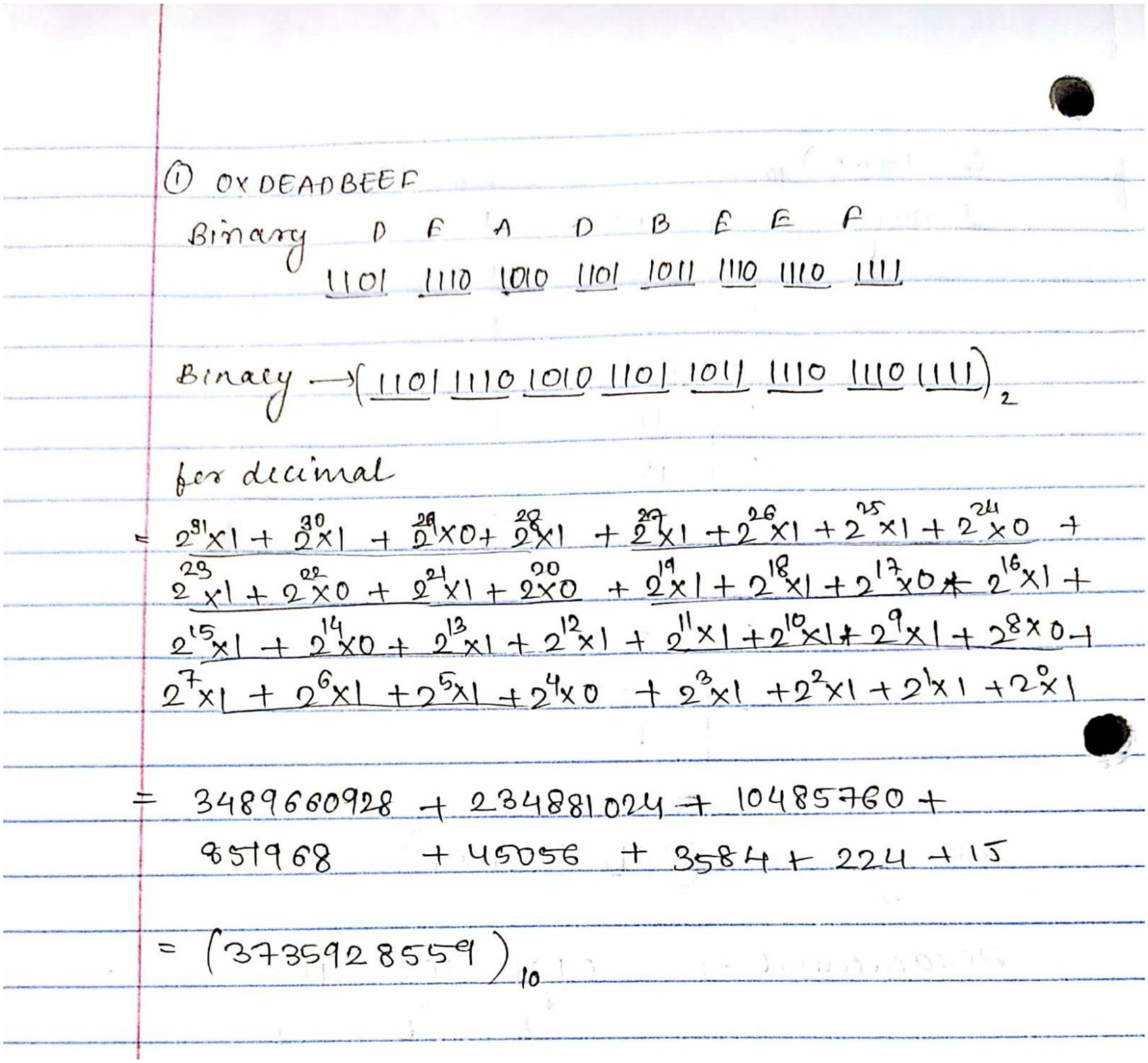
◆ Patch ◇ Tree

Comments
git-assignment/students_fa2017/students.json

[4.1]

Decimal	**Binary (show in multiples of four bits) [underscore for readability]**	**Hexadecimal**	**Suggested Data Type that can be used to hold this number**
3735928559	0b1101_1110_1010_1101_1011_1110_1110_1111	**0xDEADBEEF**	int64_t, uint64_t, uint32_t
1,465	0b0101_1011_1001	0x5B9	uint16_t, int16_t
3758	**0b1110_1010_1110**	0xEAE	int16_t, uint16_1
-172	0b0101_0100	0x54	int16_t
172	0b1010_1100	0xAC	**uint16_t,int16_t,uin8_t**

1.



① 0x DEADBEEF

Binary D E A D B E E F
1101 1110 1010 1101 1011 1110 1110 1111

Binary $\rightarrow (1101\ 1110\ 1010\ 1101\ 1011\ 1110\ 1110\ 1111)_2$

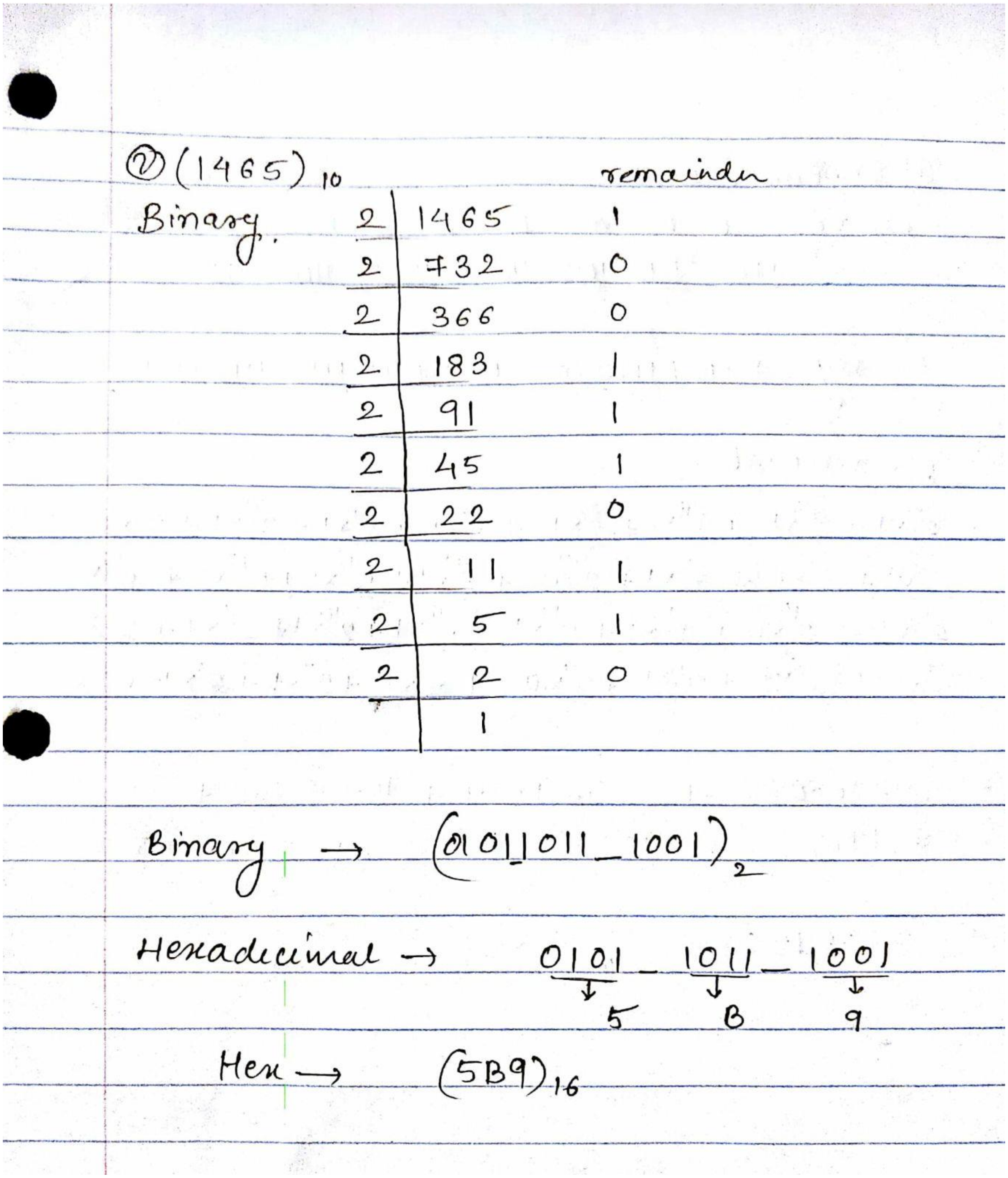
for decimal

$= 2^{31}\times1 + 2^{30}\times1 + 2^{29}\times0 + 2^{28}\times1 + 2^{27}\times1 + 2^{26}\times1 + 2^{25}\times1 + 2^{24}\times0 +$
$2^{23}\times1 + 2^{22}\times0 + 2^{21}\times1 + 2^{20}\times0 + 2^{19}\times1 + 2^{18}\times1 + 2^{17}\times0 + 2^{16}\times1 +$
$2^{15}\times1 + 2^{14}\times0 + 2^{13}\times1 + 2^{12}\times1 + 2^{11}\times1 + 2^{10}\times1 + 2^{9}\times1 + 2^{8}\times0 +$
$2^{7}\times1 + 2^{6}\times1 + 2^{5}\times1 + 2^{4}\times0 + 2^{3}\times1 + 2^{2}\times1 + 2^{1}\times1 + 2^{0}\times1$

$= 3489660928 + 234881024 + 10485760 +$
$851968 + 45056 + 3584 + 224 + 15$

$= (3735928559)_{10}$

2.



② $(1465)_{10}$

Binary.

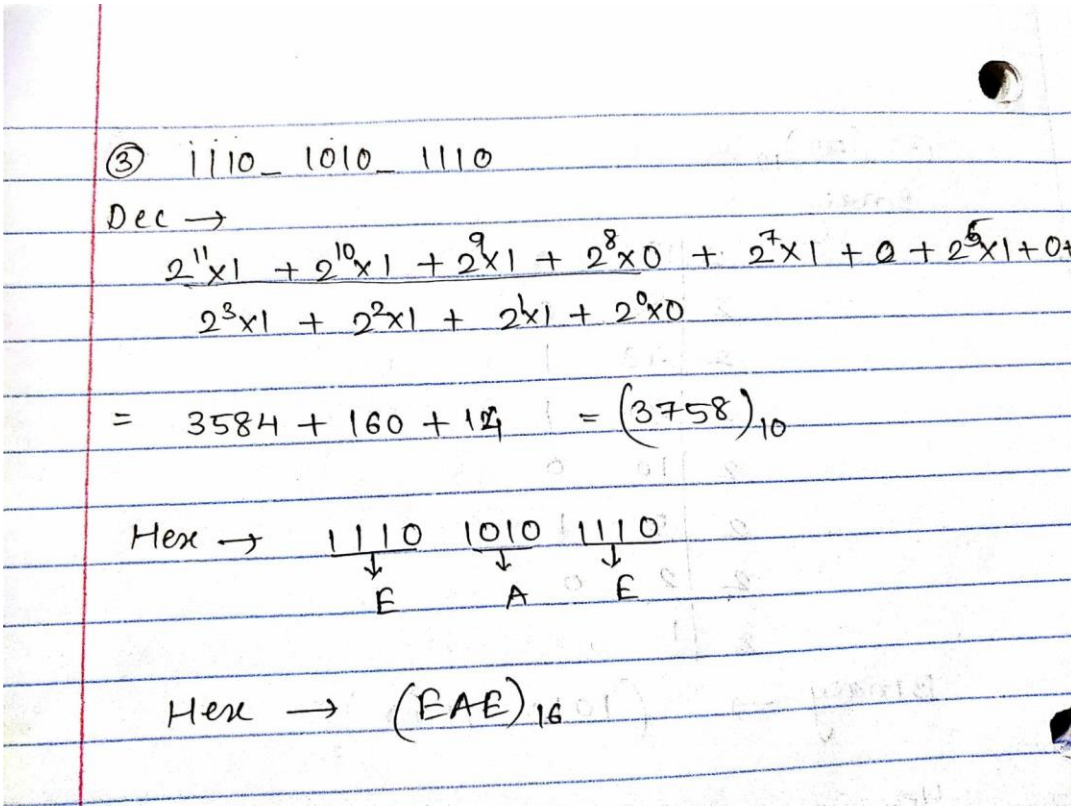
		remainder
2	1465	1
2	732	0
2	366	0
2	183	1
2	91	1
2	45	1
2	22	0
2	11	1
2	5	1
2	2	0
	1	

Binary → $(0101\_1011\_1001)_2$

Hexadecimal → $\underbrace{0101}_{\downarrow\ 5}$ _ $\underbrace{1011}_{\downarrow\ B}$ _ $\underbrace{1001}_{\downarrow\ 9}$

Hex → $(5B9)_{16}$

3.



③ 1110_1010_1110

Dec →

$$2^{11}\times1 + 2^{10}\times1 + 2^{9}\times1 + 2^{8}\times0 + 2^{7}\times1 + 0 + 2^{5}\times1 + 0 +$$
$$2^{3}\times1 + 2^{2}\times1 + 2^{1}\times1 + 2^{0}\times0$$

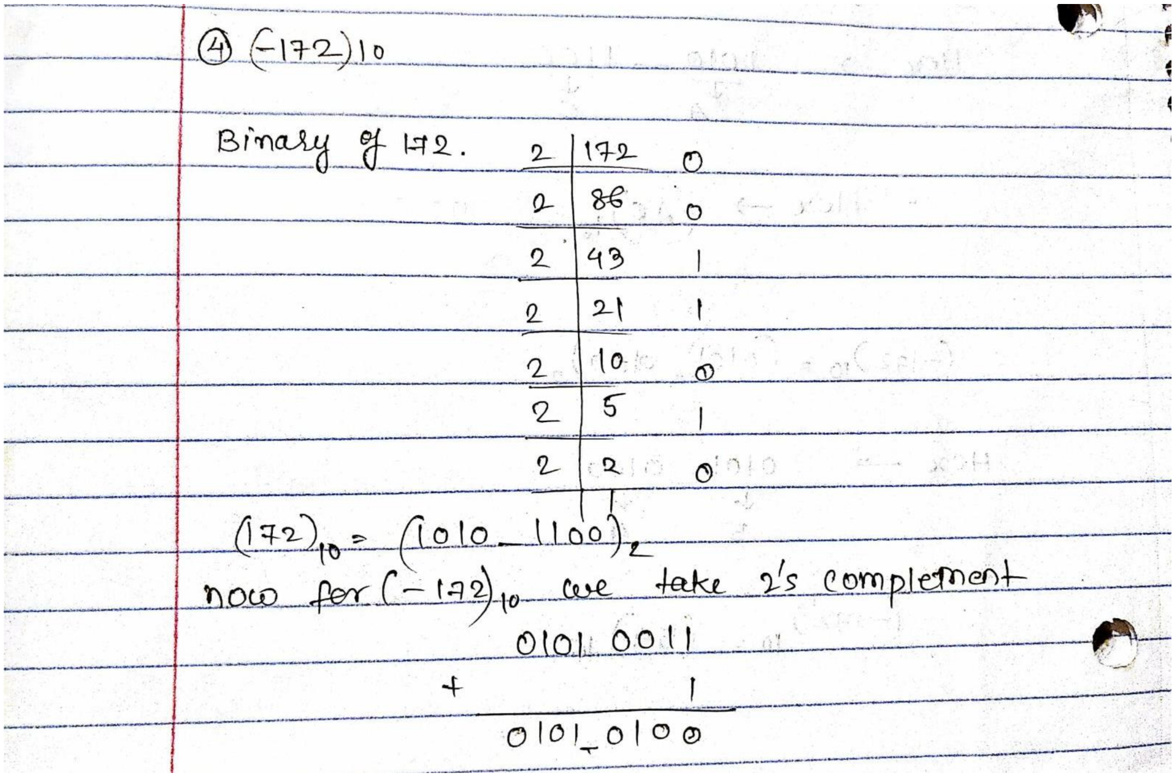
$= 3584 + 160 + 14 = (3758)_{10}$

Hex → 1110 1010 1110

↓ E ↓ A ↓ E

Hex → $(EAE)_{16}$

4.



④ $(-172)_{10}$

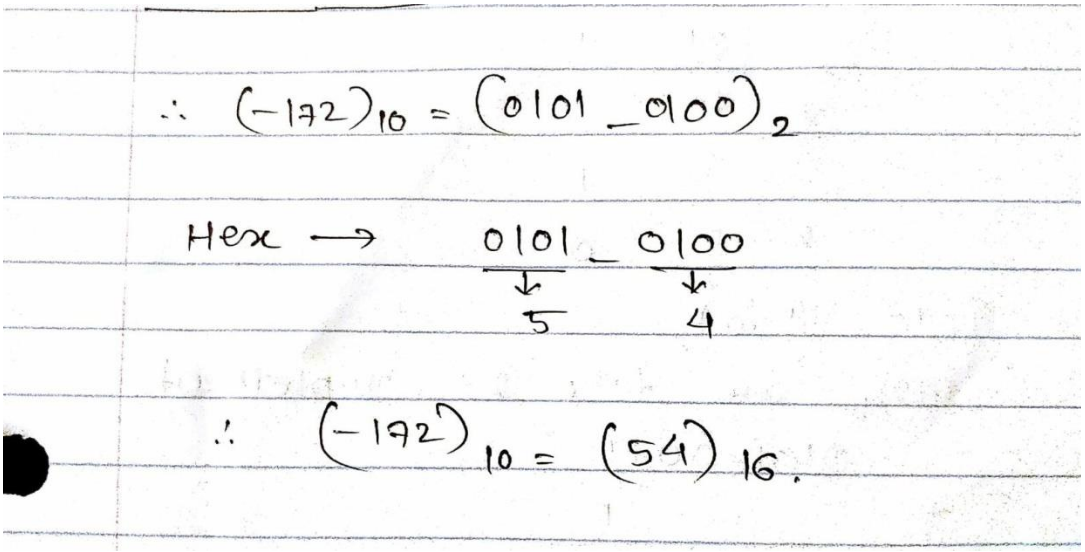
Binary of 172.

2	172	0
2	86	0
2	43	1
2	21	1
2	10	0
2	5	1
2	2	0

$(172)_{10} = (1010\_1100)_2$

now for $(-172)_{10}$ we take 2's complement

$$\begin{array}{r} 0101\ 0011 \\ +\qquad\qquad 1 \\ \hline 0101\_0100 \end{array}$$

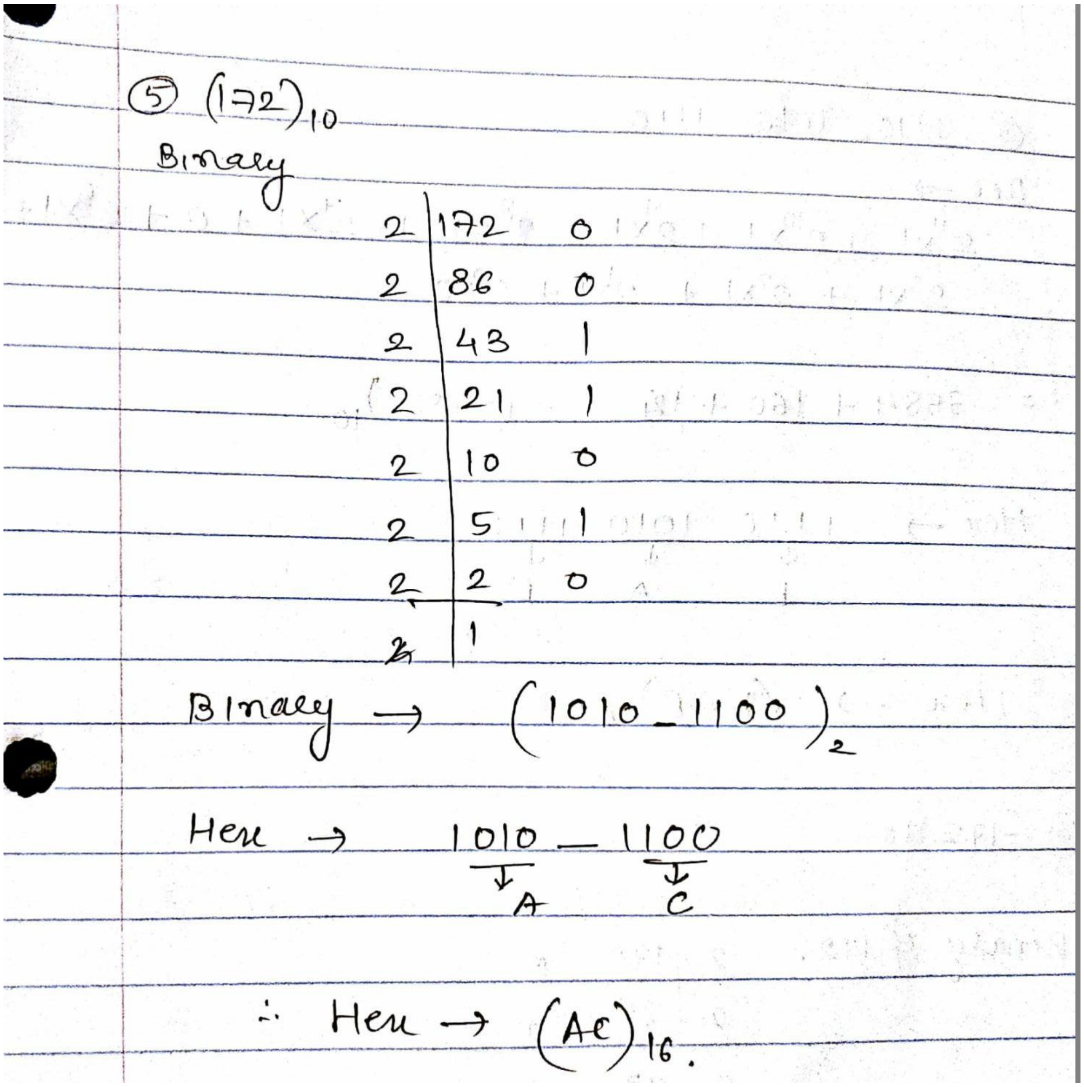


∴ $(-172)_{10} = (0101\_0100)_2$

Hex → $\underset{\downarrow \atop 5}{\underline{0101}}\_\underset{\downarrow \atop 4}{\underline{0100}}$

∴ $(-172)_{10} = (54)_{16}$.

5.



⑤ $(172)_{10}$

Binary

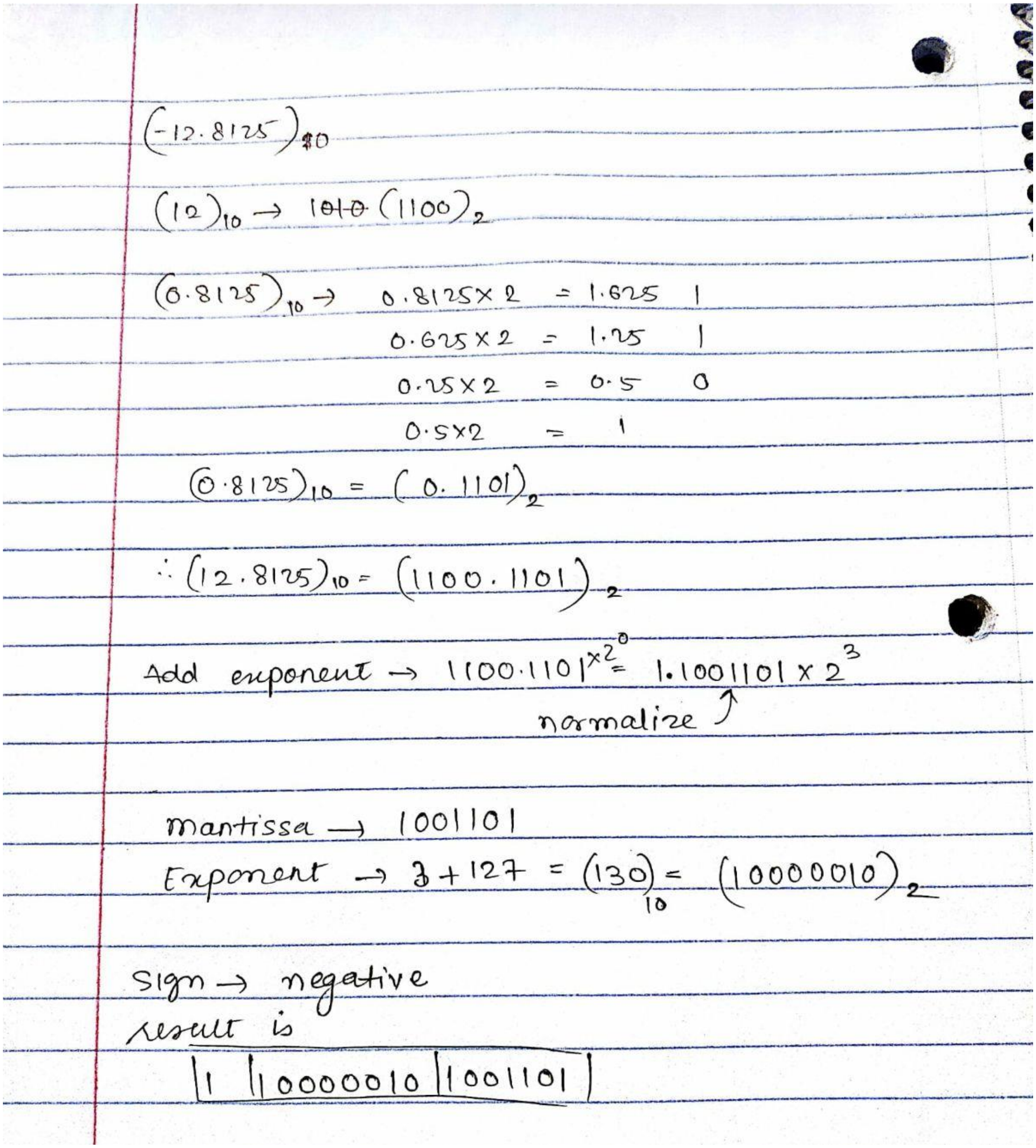
2	172	0
2	86	0
2	43	1
2	21	1
2	10	0
2	5	1
2	2	0
	1	

Binary → $(1010\_1100)_2$

Hex → $\underset{\downarrow \atop A}{\underline{1010}}\_\underset{\downarrow \atop C}{\underline{1100}}$

∴ Hex → $(AC)_{16}$.

[4.2]
32 bit floating point



$(-12.8125)_{10}$

$(12)_{10} \to$ ~~1010~~ $(1100)_2$

$(0.8125)_{10} \to$ $0.8125 \times 2 = 1.625$ 1

$0.625 \times 2 = 1.25$ 1

$0.25 \times 2 = 0.5$ 0

$0.5 \times 2 = 1$

$(0.8125)_{10} = (0.1101)_2$

$\therefore (12.8125)_{10} = (1100.1101)_2$

Add exponent $\to 1100.1101 \times 2^0 = 1.1001101 \times 2^3$

normalize

Mantissa → 1001101

Exponent $\to 3 + 127 = (130)_{10} = (10000010)_2$

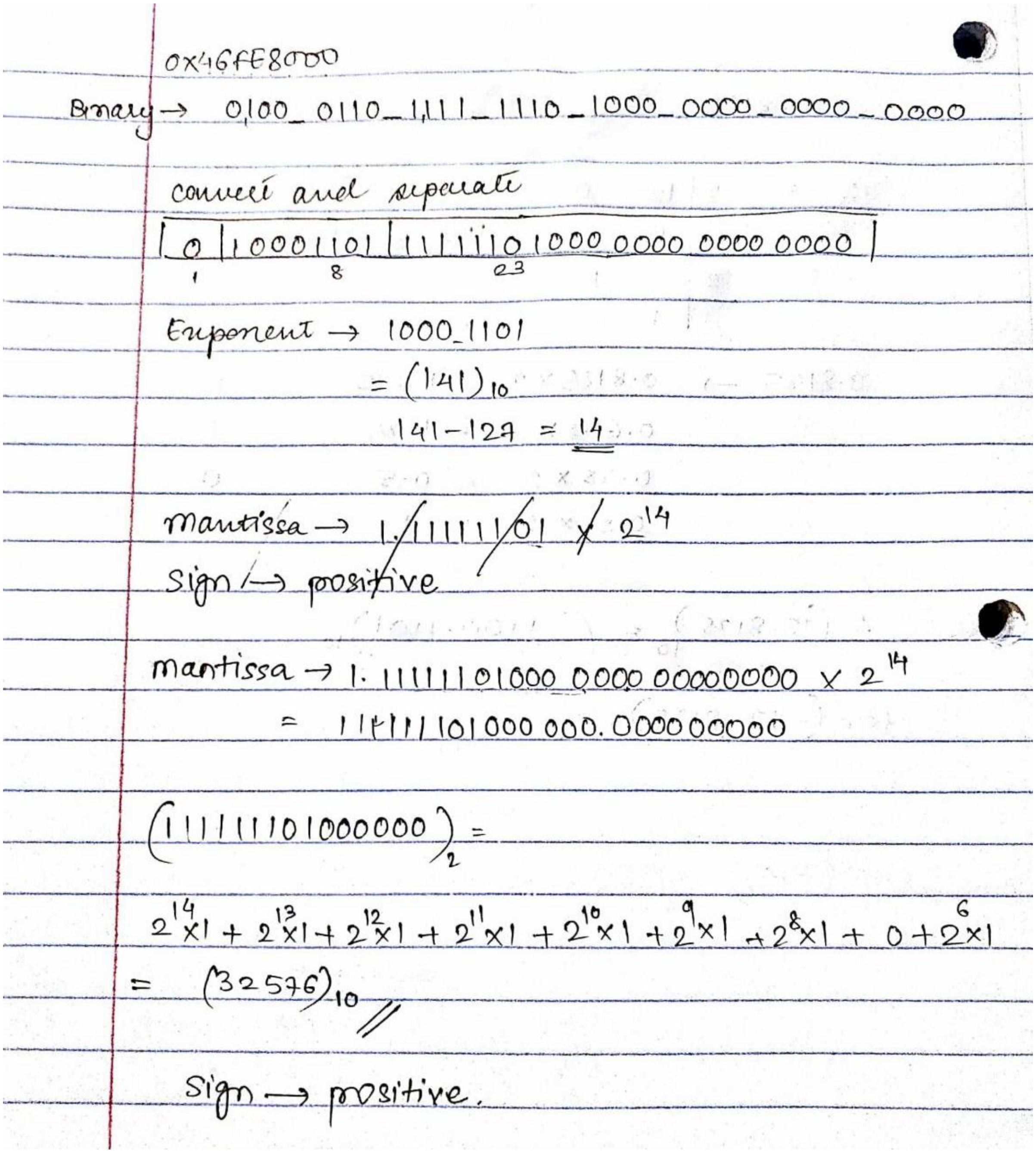
Sign → negative

result is

| 1 | 10000010 | 1001101 |

(another image missing)

[4.3]



0x46FE8000

Binary → 0100_0110_1111_1110_1000_0000_0000_0000

convert and separate

$$\underbrace{\boxed{0}}_{1}\underbrace{\boxed{10001101}}_{8}\underbrace{\boxed{1111110\,1000\;0000\;0000\;0000}}_{23}$$

Exponent → 1000_1101

$= (141)_{10}$

$141 - 127 = \underline{\underline{14}}$

~~mantissa → 1.11111101×2^{14}~~

~~Sign → positive~~

mantissa → $1.\,11111101000\;0000\;00000000 \times 2^{14}$

$= 111111101000\,000.000000000$

$(111111101000000)_2 =$

$2^{14}\times1 + 2^{13}\times1 + 2^{12}\times1 + 2^{11}\times1 + 2^{10}\times1 + 2^{9}\times1 + 2^{8}\times1 + 0 + 2^{6}\times1$

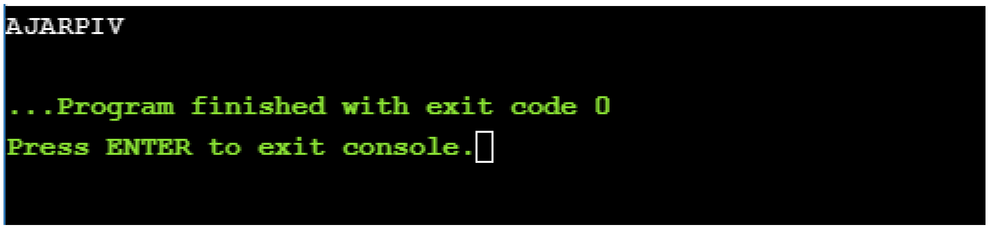
$= (32576)_{10}$ //

Sign → positive.

[5.1]
(1)

```
#include <stdio.h>
void reverse(char *str, int length);
int main()
{
    char a[] = "VIPRAJA";
    int i, length;
    char *str;
    length = getlength(a);
    str = &a[length];
    reverse(a,length);
    return 0;
}
int getlength(char *a){
    int length = 0;
    while(a[length] != '\0')
    {
        length++;
    }
    return length;
}
void reverse(char *str, int length)
{   for(int i=0; i<length; i++)
    {
    str++;
    }
    for(int i = length; i >= 0; i--)
    {
        printf("%c", *str--);
    }
}
```

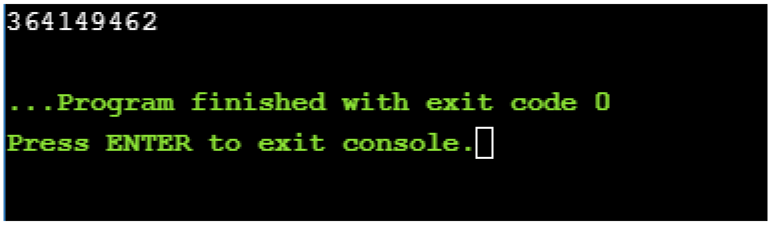
Output:



```
AJARPIV

...Program finished with exit code 0
Press ENTER to exit console.
```

(2)

```
#include <stdio.h>
void reverse(char *str, int length);
int main()
{
    char a[] = "264941463";
    int i, length;
    char *str;
    length = 4;
    length = getlength(a);
    str = &a[length];
    reverse(a,length);
    return 0;
}
int getlength(char *a){
    int length = 0;
    while(a[length] != '\0')
    {
        length++;
    }
    return length;
}
void reverse(char *str, int length)
{   for(int i=0; i<length; i++)
    {
    str++;
    }
    for(int i = length; i >= 0; i--)
    {
        printf("%c", *str--);
    }

}
```

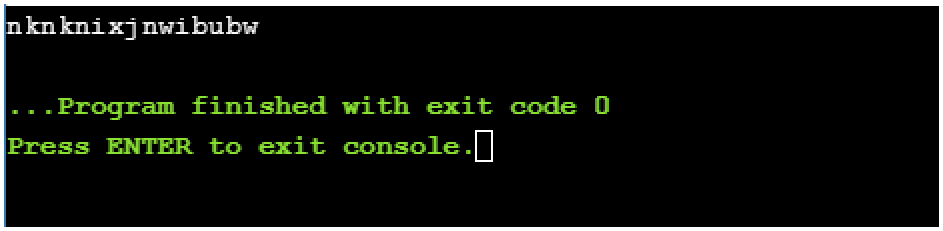


```
364149462

...Program finished with exit code 0
Press ENTER to exit console.
```

(3)

```
#include <stdio.h>
```

```
void reverse(char *str, int length);
int main()
{
    char a[] = "wbubiwnjxinknkn";
    int i, length;
    char *str;
    length = 4;
    length = getlength(a);
    str = &a[length];
    reverse(a,length);
    return 0;
}
int getlength(char *a){
    int length = 0;
    while(a[length] != '\0')
    {
        length++;
    }
    return length;
}
void reverse(char *str, int length)
{   for(int i=0; i<length; i++)
    {
    str++;
    }
    for(int i = length; i >= 0; i--)
    {
        printf("%c", *str--);
    }

}
```

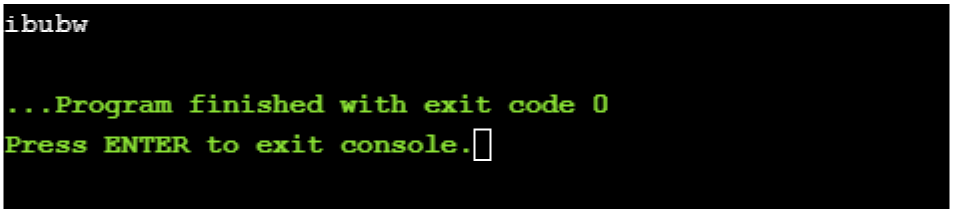


```
nknknixjnwibubw

...Program finished with exit code 0
Press ENTER to exit console.
```

Negative testing

(4)

```
#include <stdio.h>
void reverse(char *str, int length);
int main()
{
   char a[] = "wbubiwnjxinknkn";
   int i, length;
   char *str;
   length = 4;
   //length = getlength(a);
   str = &a[length];
   reverse(a,length);
   return 0;
}
int getlength(char *a){
   int length = 0;
   while(a[length] != '\0')
   {
      length++;
   }
   return length;
}
void reverse(char *str, int length)
{   for(int i=0; i<length; i++)
   {
   str++;
   }
   for(int i = length; i >= 0; i--)
   {
      printf("%c", *str--);
   }

}
```

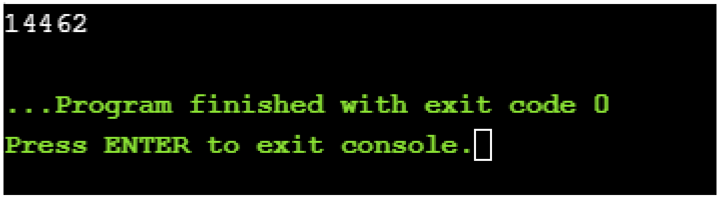


```
ibubw

...Program finished with exit code 0
Press ENTER to exit console.
```

It returns wrong reversed array

(5)

```
#include <stdio.h>
void reverse(char *str, int length);
int main()
{
   char a[] = "2644155455";
   int i, length;
   char *str;
   length = 4;
   //length = getlength(a);
   str = &a[length];
   reverse(a,length);
   return 0;
}
int getlength(char *a){
   int length = 0;
   while(a[length] != '\0')
   {
      length++;
   }
   return length;
}
void reverse(char *str, int length)
{   for(int i=0; i<length; i++)
   {
   str++;
   }
   for(int i = length; i >= 0; i--)
   {
      printf("%c", *str--);
   }

}
```



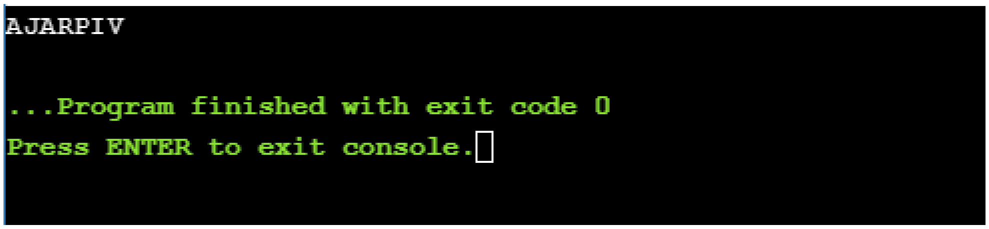
```
14462

...Program finished with exit code 0
Press ENTER to exit console.
```

It returns wrong array.

[5.2]

```
#include <stdio.h>
void reverse(char *str, int length);
int main()
{
    char a[] = "VIPRAJA";
    int i, length;
    char *str;
    length = getlength(a);
    str = &a[length];
    reverse(a,length);
    return 0;
}
int getlength(char *a){
    int length = 0;
    while(a[length] != '\0')
    {
        length++;
    }
    return length;
}
void reverse(char *str, int length)
{   for(int i=0; i<length; i++)
    {
    str++;
    }
    for(int i = length; i >= 0; i--)
    {
        printf("%c", *str--);
    }
}
```

Output:



```
AJARPIV

...Program finished with exit code 0
Press ENTER to exit console.
```