Project for Software Quality Assurance (COMP 5710)

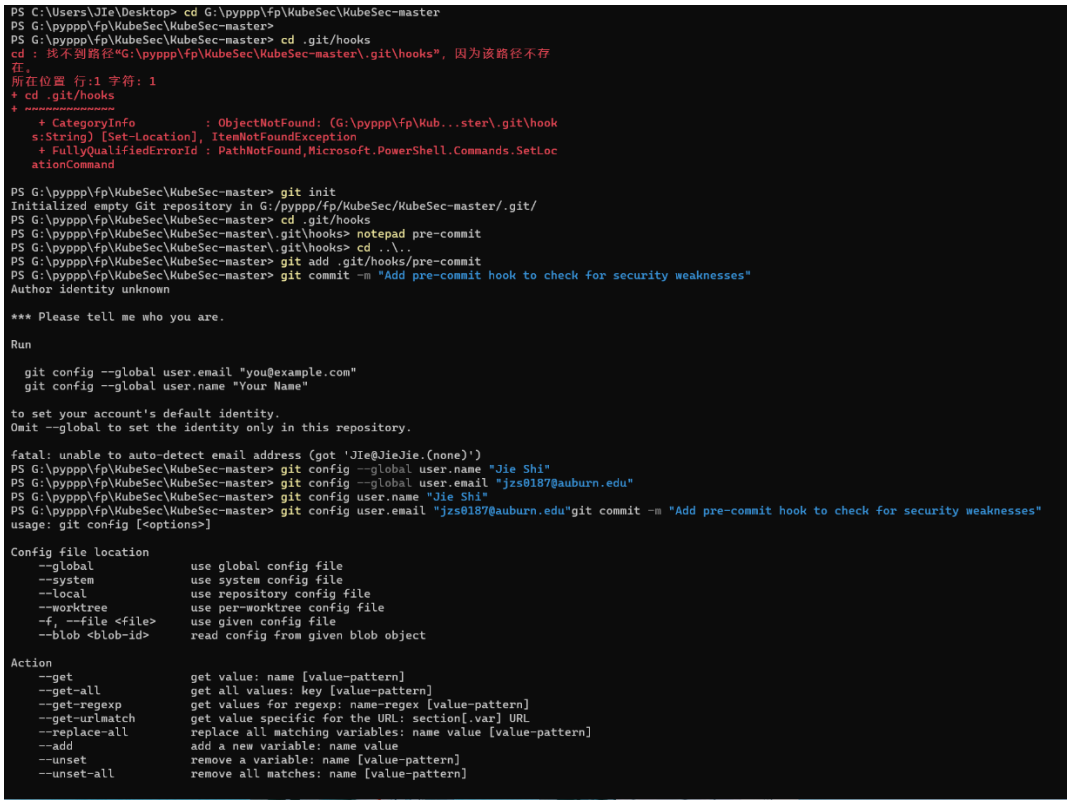
Team Members: Jie-Shi-TEAM

Team Name: Jie Shi

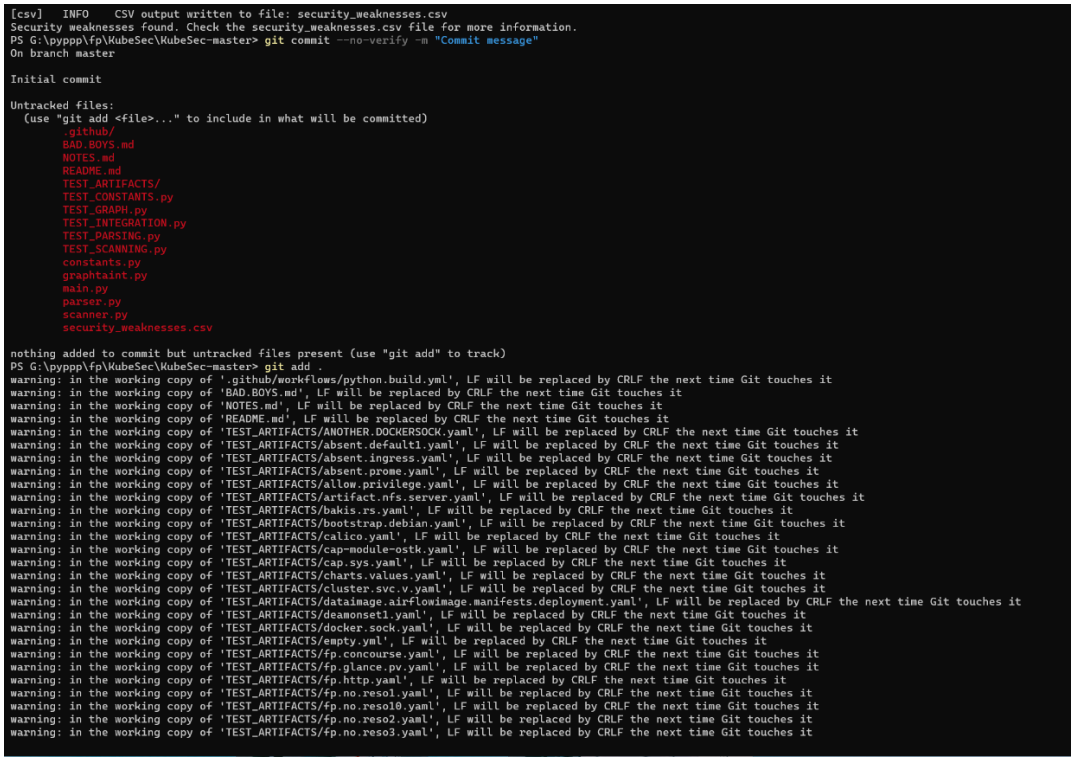
30 / April / 2023

Table of Contents

List of Figures



```
PS C:\Users\JIe\Desktop> cd G:\pyppp\fp\KubeSec\KubeSec-master
PS G:\pyppp\fp\KubeSec\KubeSec-master>
PS G:\pyppp\fp\KubeSec\KubeSec-master> cd .git/hooks
cd : 找不到路径“G:\pyppp\fp\KubeSec\KubeSec-master\.git\hooks”，因为该路径不存
在。
所在位置 行:1 字符: 1
+ cd .git/hooks
+ ~~~~~~~~~~~~~
    + CategoryInfo          : ObjectNotFound: (G:\pyppp\fp\Kub...ster\.git\hook
   s:String) [Set-Location], ItemNotFoundException
    + FullyQualifiedErrorId : PathNotFound,Microsoft.PowerShell.Commands.SetLoc
   ationCommand

PS G:\pyppp\fp\KubeSec\KubeSec-master> git init
Initialized empty Git repository in G:/pyppp/fp/KubeSec/KubeSec-master/.git/
PS G:\pyppp\fp\KubeSec\KubeSec-master> cd .git/hooks
PS G:\pyppp\fp\KubeSec\KubeSec-master\.git\hooks> notepad pre-commit
PS G:\pyppp\fp\KubeSec\KubeSec-master\.git\hooks> cd ..\..
PS G:\pyppp\fp\KubeSec\KubeSec-master> git add .git/hooks/pre-commit
PS G:\pyppp\fp\KubeSec\KubeSec-master> git commit -m "Add pre-commit hook to check for security weaknesses"
Author identity unknown

*** Please tell me who you are.

Run

  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'JIe@JieJie.(none)')
PS G:\pyppp\fp\KubeSec\KubeSec-master> git config --global user.name "Jie Shi"
PS G:\pyppp\fp\KubeSec\KubeSec-master> git config --global user.email "jzs0187@auburn.edu"
PS G:\pyppp\fp\KubeSec\KubeSec-master> git config user.name "Jie Shi"
PS G:\pyppp\fp\KubeSec\KubeSec-master> git config user.email "jzs0187@auburn.edu"git commit -m "Add pre-commit hook to check for security weaknesses"
usage: git config [<options>]

Config file location
    --global              use global config file
    --system              use system config file
    --local               use repository config file
    --worktree            use per-worktree config file
    -f, --file <file>     use given config file
    --blob <blob-id>      read config from given blob object

Action
    --get                 get value: name [value-pattern]
    --get-all             get all values: key [value-pattern]
    --get-regexp          get values for regexp: name-regex [value-pattern]
    --get-urlmatch        get value specific for the URL: section[.var] URL
    --replace-all         replace all matching variables: name value [value-pattern]
    --add                 add a new variable: name value
    --unset               remove a variable: name [value-pattern]
    --unset-all           remove all matches: name [value-pattern]
```



```
[csv]   INFO    CSV output written to file: security_weaknesses.csv
Security weaknesses found. Check the security_weaknesses.csv file for more information.
PS G:\pyppp\fp\KubeSec\KubeSec-master> git commit --no-verify -m "Commit message"
On branch master

Initial commit

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .github/
        BAD.BOYS.md
        NOTES.md
        README.md
        TEST_ARTIFACTS/
        TEST_CONSTANTS.py
        TEST_GRAPH.py
        TEST_INTEGRATION.py
        TEST_PARSING.py
        TEST_SCANNING.py
        constants.py
        graphtaint.py
        main.py
        parser.py
        scanner.py
        security_weaknesses.csv

nothing added to commit but untracked files present (use "git add" to track)
PS G:\pyppp\fp\KubeSec\KubeSec-master> git add .
warning: in the working copy of '.github/workflows/python.build.yml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'BAD.BOYS.md', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'NOTES.md', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'README.md', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'TEST_ARTIFACTS/ANOTHER.DOCKERSOCK.yaml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'TEST_ARTIFACTS/absent.default1.yaml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'TEST_ARTIFACTS/absent.ingress.yaml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'TEST_ARTIFACTS/absent.prome.yaml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'TEST_ARTIFACTS/allow.privilege.yaml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'TEST_ARTIFACTS/artifact.nfs.server.yaml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'TEST_ARTIFACTS/bakis.rs.yaml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'TEST_ARTIFACTS/bootstrap.debian.yaml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'TEST_ARTIFACTS/calico.yaml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'TEST_ARTIFACTS/cap-module-ostk.yaml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'TEST_ARTIFACTS/cap.sys.yaml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'TEST_ARTIFACTS/charts.values.yaml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'TEST_ARTIFACTS/cluster.svc.v.yaml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'TEST_ARTIFACTS/dataimage.airflowimage.manifests.deployment.yaml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'TEST_ARTIFACTS/deamonset1.yaml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'TEST_ARTIFACTS/docker.sock.yaml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'TEST_ARTIFACTS/empty.yml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'TEST_ARTIFACTS/fp.concourse.yaml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'TEST_ARTIFACTS/fp.glance.pv.yaml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'TEST_ARTIFACTS/fp.http.yaml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'TEST_ARTIFACTS/fp.no.reso1.yaml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'TEST_ARTIFACTS/fp.no.reso10.yaml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'TEST_ARTIFACTS/fp.no.reso2.yaml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'TEST_ARTIFACTS/fp.no.reso3.yaml', LF will be replaced by CRLF the next time Git touches it
```

```
 create mode 100644 TEST_ARTIFACTS/roll.present.deploy.yaml
 create mode 100644 TEST_ARTIFACTS/sample-nfs-server.yaml
 create mode 100644 TEST_ARTIFACTS/skampi.values.yaml
 create mode 100644 TEST_ARTIFACTS/sni.values.yaml
 create mode 100644 TEST_ARTIFACTS/special.secret1.yaml
 create mode 100644 TEST_ARTIFACTS/start_vault_temp.yaml
 create mode 100644 TEST_ARTIFACTS/tango.values.yaml
 create mode 100644 TEST_ARTIFACTS/tasks.main.yml
 create mode 100644 TEST_ARTIFACTS/test.sample.pod.yaml
 create mode 100644 TEST_ARTIFACTS/tp.default.nspace.yaml
 create mode 100644 TEST_ARTIFACTS/tp.host.net2.yaml
 create mode 100644 TEST_ARTIFACTS/tp.nsp.dflt.yaml
 create mode 100644 TEST_ARTIFACTS/tp.seccomp.unconfined.yaml
 create mode 100644 TEST_ARTIFACTS/tp_secu_context_miss.yaml
 create mode 100644 TEST_CONSTANTS.py
 create mode 100644 TEST_GRAPH.py
 create mode 100644 TEST_INTEGRATION.py
 create mode 100644 TEST_PARSING.py
 create mode 100644 TEST_SCANNING.py
 create mode 100644 constants.py
 create mode 100644 graphtaint.py
 create mode 100644 main.py
 create mode 100644 parser.py
 create mode 100644 scanner.py
 create mode 100644 security_weaknesses.csv
PS G:\pyppp\fp\KubeSec\KubeSec-master> git log
commit e75ef3eefb33c276303a95395ebb82b5b15fd7e7 (HEAD -> master)
Author: Jie Shi <jzs0187@auburn.edu>
Date:   Fri Apr 21 02:29:04 2023 -0500

    Initial commit with pre-commit hook and project files
PS G:\pyppp\fp\KubeSec\KubeSec-master>
```

Figure 2: Output of ‘fuzz.py’ in GitHub actions

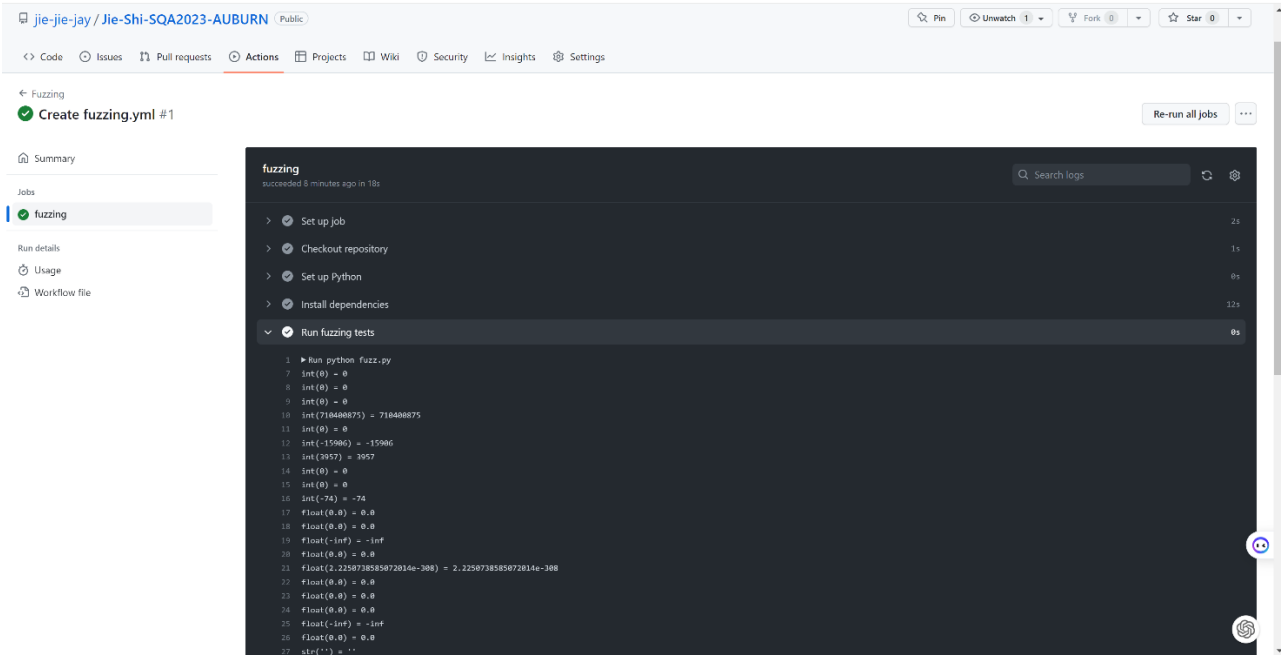


Figure 3: Example of a successful GitHub action

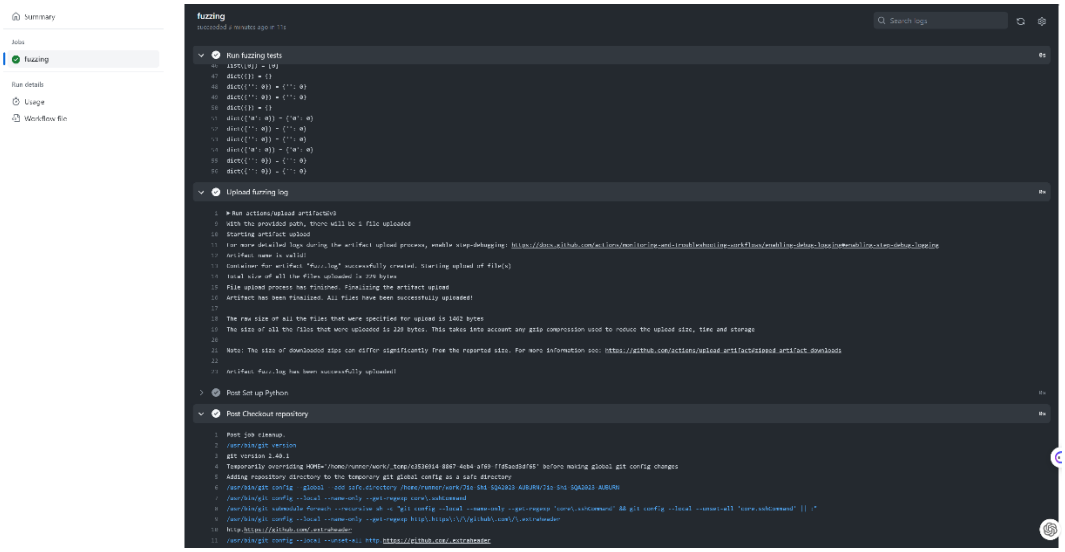
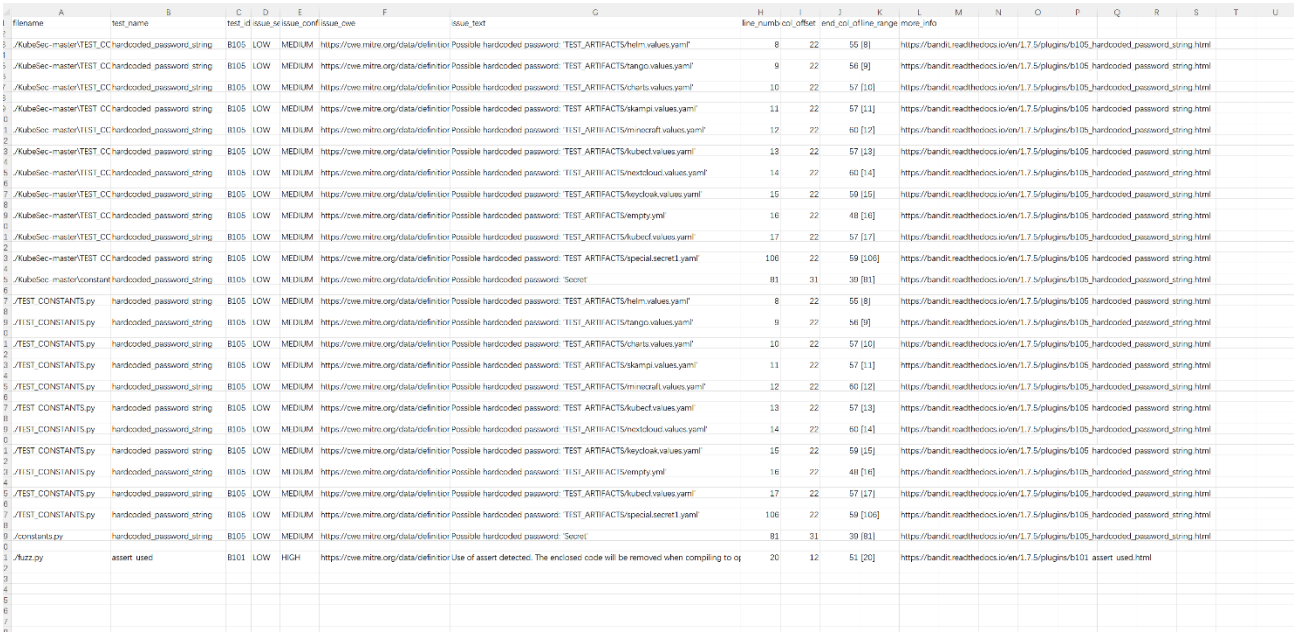


Figure 4: Logging commands

```python
import logging
from hypothesis import given, settings
from hypothesis.strategies import integers, floats, text, lists, dictionaries

logging.basicConfig(filename='fuzz.log', level=logging.DEBUG)

# Create an empty fuzz.log file if it doesn't exist
with open('fuzz.log', 'a') as f:
    pass

test_cases = [
    ('int', int, integers()),
    ('float', float, floats(allow_nan=False)),
    ('str', str, text()),
    ('list', list, lists(integers())),
    ('dict', dict, dictionaries(text(), integers()))
]

Jie Shi *
def run_tests():
    for func_name, func, strategy in test_cases:
        Jie Shi *
        @settings(max_examples=10)
        @given(strategy)
        def test(value):
            result = func(value)
            assert isinstance(result, type(func()))

            log_message = f"{func_name}({value!r}) = {result!r}"
            print(log_message)
            logging.debug(log_message)
        test()

if __name__ == "__main__":
    run_tests()
```

List of Tables

Table 1: Output of security weaknesses from git hook



filename	test_name	test_id	issue_se	issue_conf	issue_cwe	issue_text	line_numb	col_offset	end_col_of	line_range	more_info
./KubeSec-master\TEST_CC	hardcoded_password_string	B105	LOW	MEDIUM	https://cwe.mitre.org/data/definitior	Possible hardcoded password: 'TEST_ARTIFACTS/helm.values.yaml'	8	22	55	[8]	https://bandit.readthedocs.io/en/1.7.5/plugins/b105_hardcoded_password_string.html
./KubeSec-master\TEST_CC	hardcoded_password_string	B105	LOW	MEDIUM	https://cwe.mitre.org/data/definitior	Possible hardcoded password: 'TEST_ARTIFACTS/tango.values.yaml'	9	22	56	[9]	https://bandit.readthedocs.io/en/1.7.5/plugins/b105_hardcoded_password_string.html
./KubeSec-master\TEST_CC	hardcoded_password_string	B105	LOW	MEDIUM	https://cwe.mitre.org/data/definitior	Possible hardcoded password: 'TEST_ARTIFACTS/charts.values.yaml'	10	22	57	[10]	https://bandit.readthedocs.io/en/1.7.5/plugins/b105_hardcoded_password_string.html
./KubeSec-master\TEST_CC	hardcoded_password_string	B105	LOW	MEDIUM	https://cwe.mitre.org/data/definitior	Possible hardcoded password: 'TEST_ARTIFACTS/skampi.values.yaml'	11	22	57	[11]	https://bandit.readthedocs.io/en/1.7.5/plugins/b105_hardcoded_password_string.html
./KubeSec-master\TEST_CC	hardcoded_password_string	B105	LOW	MEDIUM	https://cwe.mitre.org/data/definitior	Possible hardcoded password: 'TEST_ARTIFACTS/minecraft.values.yaml'	12	22	60	[12]	https://bandit.readthedocs.io/en/1.7.5/plugins/b105_hardcoded_password_string.html
./KubeSec-master\TEST_CC	hardcoded_password_string	B105	LOW	MEDIUM	https://cwe.mitre.org/data/definitior	Possible hardcoded password: 'TEST_ARTIFACTS/kubecf.values.yaml'	13	22	57	[13]	https://bandit.readthedocs.io/en/1.7.5/plugins/b105_hardcoded_password_string.html
./KubeSec-master\TEST_CC	hardcoded_password_string	B105	LOW	MEDIUM	https://cwe.mitre.org/data/definitior	Possible hardcoded password: 'TEST_ARTIFACTS/mextcloud.values.yaml'	14	22	60	[14]	https://bandit.readthedocs.io/en/1.7.5/plugins/b105_hardcoded_password_string.html
./KubeSec-master\TEST_CC	hardcoded_password_string	B105	LOW	MEDIUM	https://cwe.mitre.org/data/definitior	Possible hardcoded password: 'TEST_ARTIFACTS/keycloak.values.yaml'	15	22	59	[15]	https://bandit.readthedocs.io/en/1.7.5/plugins/b105_hardcoded_password_string.html
./KubeSec-master\TEST_CC	hardcoded_password_string	B105	LOW	MEDIUM	https://cwe.mitre.org/data/definitior	Possible hardcoded password: 'TEST_ARTIFACTS/empty.yml'	16	22	48	[16]	https://bandit.readthedocs.io/en/1.7.5/plugins/b105_hardcoded_password_string.html
./KubeSec-master\TEST_CC	hardcoded_password_string	B105	LOW	MEDIUM	https://cwe.mitre.org/data/definitior	Possible hardcoded password: 'TEST_ARTIFACTS/kubecf.values.yaml'	17	22	57	[17]	https://bandit.readthedocs.io/en/1.7.5/plugins/b105_hardcoded_password_string.html
./KubeSec-master\TEST_CC	hardcoded_password_string	B105	LOW	MEDIUM	https://cwe.mitre.org/data/definitior	Possible hardcoded password: 'TEST_ARTIFACTS/special.secret1.yaml'	106	22	59	[106]	https://bandit.readthedocs.io/en/1.7.5/plugins/b105_hardcoded_password_string.html
./KubeSec-master\constant	hardcoded_password_string	B105	LOW	MEDIUM	https://cwe.mitre.org/data/definitior	Possible hardcoded password: 'Secret'	81	31	39	[81]	https://bandit.readthedocs.io/en/1.7.5/plugins/b105_hardcoded_password_string.html
./TEST_CONSTANTS.py	hardcoded_password_string	B105	LOW	MEDIUM	https://cwe.mitre.org/data/definitior	Possible hardcoded password: 'TEST_ARTIFACTS/helm.values.yaml'	8	22	55	[8]	https://bandit.readthedocs.io/en/1.7.5/plugins/b105_hardcoded_password_string.html
./TEST_CONSTANTS.py	hardcoded_password_string	B105	LOW	MEDIUM	https://cwe.mitre.org/data/definitior	Possible hardcoded password: 'TEST_ARTIFACTS/tango.values.yaml'	9	22	56	[9]	https://bandit.readthedocs.io/en/1.7.5/plugins/b105_hardcoded_password_string.html
./TEST_CONSTANTS.py	hardcoded_password_string	B105	LOW	MEDIUM	https://cwe.mitre.org/data/definitior	Possible hardcoded password: 'TEST_ARTIFACTS/charts.values.yaml'	10	22	57	[10]	https://bandit.readthedocs.io/en/1.7.5/plugins/b105_hardcoded_password_string.html
./TEST_CONSTANTS.py	hardcoded_password_string	B105	LOW	MEDIUM	https://cwe.mitre.org/data/definitior	Possible hardcoded password: 'TEST_ARTIFACTS/skampi.values.yaml'	11	22	57	[11]	https://bandit.readthedocs.io/en/1.7.5/plugins/b105_hardcoded_password_string.html
./TEST_CONSTANTS.py	hardcoded_password_string	B105	LOW	MEDIUM	https://cwe.mitre.org/data/definitior	Possible hardcoded password: 'TEST_ARTIFACTS/minecraft.values.yaml'	12	22	60	[12]	https://bandit.readthedocs.io/en/1.7.5/plugins/b105_hardcoded_password_string.html
./TEST_CONSTANTS.py	hardcoded_password_string	B105	LOW	MEDIUM	https://cwe.mitre.org/data/definitior	Possible hardcoded password: 'TEST_ARTIFACTS/kubecf.values.yaml'	13	22	57	[13]	https://bandit.readthedocs.io/en/1.7.5/plugins/b105_hardcoded_password_string.html
./TEST_CONSTANTS.py	hardcoded_password_string	B105	LOW	MEDIUM	https://cwe.mitre.org/data/definitior	Possible hardcoded password: 'TEST_ARTIFACTS/mextcloud.values.yaml'	14	22	60	[14]	https://bandit.readthedocs.io/en/1.7.5/plugins/b105_hardcoded_password_string.html
./TEST_CONSTANTS.py	hardcoded_password_string	B105	LOW	MEDIUM	https://cwe.mitre.org/data/definitior	Possible hardcoded password: 'TEST_ARTIFACTS/keycloak.values.yaml'	15	22	59	[15]	https://bandit.readthedocs.io/en/1.7.5/plugins/b105_hardcoded_password_string.html
./TEST_CONSTANTS.py	hardcoded_password_string	B105	LOW	MEDIUM	https://cwe.mitre.org/data/definitior	Possible hardcoded password: 'TEST_ARTIFACTS/empty.yml'	16	22	48	[16]	https://bandit.readthedocs.io/en/1.7.5/plugins/b105_hardcoded_password_string.html
./TEST_CONSTANTS.py	hardcoded_password_string	B105	LOW	MEDIUM	https://cwe.mitre.org/data/definitior	Possible hardcoded password: 'TEST_ARTIFACTS/kubecf.values.yaml'	17	22	57	[17]	https://bandit.readthedocs.io/en/1.7.5/plugins/b105_hardcoded_password_string.html
./TEST_CONSTANTS.py	hardcoded_password_string	B105	LOW	MEDIUM	https://cwe.mitre.org/data/definitior	Possible hardcoded password: 'TEST_ARTIFACTS/special.secret1.yaml'	106	22	59	[106]	https://bandit.readthedocs.io/en/1.7.5/plugins/b105_hardcoded_password_string.html
./constants.py	hardcoded_password_string	B105	LOW	MEDIUM	https://cwe.mitre.org/data/definitior	Possible hardcoded password: 'Secret'	81	31	39	[81]	https://bandit.readthedocs.io/en/1.7.5/plugins/b105_hardcoded_password_string.html
./fuzz.py	assert_used	B101	LOW	HIGH	https://cwe.mitre.org/data/definitior	Use of assert detected. The enclosed code will be removed when compiling to op	20	12	51	[20]	https://bandit.readthedocs.io/en/1.7.5/plugins/b101_assert_used.html

1 Part 4a – Git Hook

I started by installing the "Bandit" security linter and creating a pre-commit hook to report any security weaknesses in my code. This was done by navigating to the ".git/hooks" directory in my project folder, creating a new file named "pre-commit", and adding the necessary script. The script checks for security weaknesses using the Bandit tool and generates a CSV file with the results. If no weaknesses are found, the script outputs a message indicating this. If weaknesses are found, the script outputs a message with instructions on how to access the CSV file for more information.

2 Part 4b – Fuzz.py

I created a fuzz.py script to perform fuzz testing on the selected Python methods. The script used the Hypothesis library to generate various test inputs and applied the methods to those inputs. I utilized the given, settings, and different strategies from Hypothesis to create test cases for each method. I implemented logging in the fuzz.py script to record the inputs and outputs of each test case. This allowed me to better analyze any issues discovered during the fuzzing process. I integrated the fuzz testing script with GitHub Actions by creating a workflow file named fuzzing.yml. This workflow was configured to run whenever changes were pushed to the new-branch or when a pull request targeting the new-branch was created. And I updated the Node.js version used by GitHub Actions to resolve a warning regarding deprecated Node.js 12 actions. And I monitored the GitHub Actions results and reviewed the logs to identify any bugs or issues in the tested Python methods.

Learned:

1. I learned how to create fuzz testing scripts using the Hypothesis library and how to apply it to various Python methods.
2. I learned how to implement logging in my fuzz testing script to record and analyze test inputs and outputs effectively.
3. I gained experience in integrating fuzz testing with GitHub Actions, allowing me to automate the testing process and ensure that my code is robust and bug-free.
4. I learned how to troubleshoot and resolve warnings in GitHub Actions, such as updating the Node.js version used by the environment.

3 Part 4c – Forensics

I started with a fuzz.py script that performed fuzz testing on the selected Python methods using the Hypothesis library. This script generated various test inputs and applied the methods to those inputs. To integrate forensics, I added logging functionality to the fuzz.py script. By importing the logging module and configuring it, I enabled the script to log the input and output data for each test case. This allowed me to collect more information about the tested functions and better analyze any potential issues discovered during the fuzzing process. I updated the fuzz.py script to include logging messages for each test case, ensuring that I captured essential details for forensics analysis. I integrated the modified fuzz testing script with GitHub Actions by updating the workflow file named fuzzing.yml. I added an additional step to upload the generated log file (fuzz.log) as an artifact, making it available for analysis and review after each GitHub Actions run. And I monitored the GitHub Actions results, downloaded the log artifacts, and reviewed the logs to identify any bugs or issues in the tested Python methods.

Learned:

1. I learned how to integrate forensics into fuzz testing by adding logging functionality to my fuzz testing script. This allowed me to collect valuable information about the inputs and outputs of the tested functions, enabling better analysis and understanding of potential issues.
2. I learned how to modify the GitHub Actions workflow file to include an additional step for uploading the generated log file as an artifact. This made it easy to access and analyze the log data after each fuzz testing run.
3. I gained experience in reviewing and analyzing log data generated during fuzz testing, helping me to identify bugs or issues in the tested Python methods more effectively.
4. I learned about the importance of integrating forensics into the software testing process. By collecting and analyzing log data, I can better understand the behavior of the tested functions, allowing me to improve the overall quality and reliability of my code.

4 Review

Despite the fact that I had previous experience with many of the activities from the workshops throughout the semester, I still managed to learn new things during this project. The most difficult part I faced was dealing with GitHub Actions. Even though I already had a basic understanding of them, getting acquainted with the inner workings of the workflow to ensure my fuzzer would run automatically proved to be a challenging process that required some experimentation.

To view my repository, simply visit the following link:
https://github.com/jie-jie-jay/Jie-Shi-SQA2023-AUBURN

Appendix A: Full Output of fuzz.py

```
PS C:\Users\JIe\Desktop> cd G:\pyppp\fp\KubeSec\KubeSec-master
PS G:\pyppp\fp\KubeSec\KubeSec-master> python fuzz.py
int(0) = 0
int(0) = 0
int(0) = 0
int(0) = 0
int(0) = 0
int(11479) = 11479
int(-1621808158) = -1621808158
int(-351754287) = -351754287
int(-63) = -63
int(0) = 0
int(-21870) = -21870
int(4683370050971417985) = 4683370050971417985
int(13771840299194870704084281704234665 2620) =
13771840299194870704084281704234665 2620
int(-231) = -231
int(0) = 0
int(-518) = -518
int(2293073786150886200) = 2293073786150886200
int(44986116496115867) = 44986116496115867
int(-12032) = -12032
int(47) = 47
int(64) = 64
int(-39) = -39
```

int(1701449803) = 1701449803

int(-20651) = -20651

int(32420) = 32420

int(30220) = 30220

int(114) = 114

int(-1436977148615391629906026852801304606 80) = -1436977148615391629906026852801304606 80

int(-640) = -640

int(-5606) = -5606

int(-5092) = -5092

int(-22809) = -22809

int(-710535993995484995) = -710535993995484995

int(15172) = 15172

int(59) = 59

int(-27548) = -27548

int(-107) = -107

int(11012) = 11012

int(43) = 43

int(-3) = -3

int(-2692) = -2692

int(58) = 58

int(4049) = 4049

int(-15) = -15

int(-481385475830149394) = -481385475830149394

int(50) = 50

int(85) = 85

int(648) = 648

int(18814) = 18814

int(73) = 73

int(-1947988039) = -1947988039

int(-1587273794) = -1587273794

int(30358) = 30358

int(-675755084633255588732129639401775662 17) = -
675755084633255588732129639401775662 17

int(-24780) = -24780

int(127) = 127

int(-1568248929882984426) = -1568248929882984426

int(24886) = 24886

int(97) = 97

int(-118) = -118

int(443576630618041724935076249110995641 82) =
443576630618041724935076249110995641 82

int(140855189855558653011320974455440894854) =
140855189855558653011320974455440894854

int(92) = 92

int(-21643621472860102 44) = -21643621472860102 44

int(72864610070538388 16) = 72864610070538388 16

int(29930) = 29930

int(16516) = 16516

int(-64) = -64

int(32297) = 32297

int(126) = 126

int(110410664471886447546630106737348152 56) =
110410664471886447546630106737348152 56

int(13642) = 13642

int(-15341) = -15341

int(6430080274755201038) = 6430080274755201038

int(-24621) = -24621

int(-872) = -872

int(3) = 3

int(-22555) = -22555

int(6802402722131413182) = 6802402722131413182

int(-70) = -70

int(-813930762) = -813930762

int(-115) = -115

int(354) = 354

int(1) = 1

int(-250290126) = -250290126

int(-30158) = -30158

int(-117) = -117

int(19720) = 19720

int(77) = 77

int(22782) = 22782

int(-88) = -88

int(98) = 98

int(-54) = -54

int(-4335) = -4335

int(-16) = -16

int(-23352) = -23352

int(-1429114588296165841318235133433184164l) = -
1429114588296165841318235133433184164l

int(4391) = 4391

int(121) = 121

int(7) = 7

float(0.0) = 0.0

float(0.0) = 0.0

float(2.2250738585072014e-308) = 2.2250738585072014e-308

float(0.0) = 0.0

float(-3.436942122986543e-224) = -3.436942122986543e-224

float(0.0) = 0.0

float(-1.1754943508222875e-38) = -1.1754943508222875e-38

float(0.0) = 0.0

float(-6.103515625e-05) = -6.103515625e-05

float(0.0) = 0.0

float(0.0) = 0.0

float(1.1532269453862838e+178) = 1.1532269453862838e+178

float(-181379763491987.0) = -181379763491987.0

float(181379763491987.0) = 181379763491987.0

float(1.1532269453850788e+178) = 1.1532269453850788e+178

float(2.273060334455596e+64) = 2.273060334455596e+64

float(4.974860208809916e+242) = 4.974860208809916e+242

float(-1.175494351e-38) = -1.175494351e-38

float(0.0) = 0.0

float(inf) = inf

float(0.0) = 0.0

float(-1.9) = -1.9

float(inf) = inf

float(0.0) = 0.0

float(-6.103515625e-05) = -6.103515625e-05

float(0.0) = 0.0

float(inf) = inf

float(0.0) = 0.0

float(2.00001) = 2.00001

float(-3.843902540445236e+16) = -3.843902540445236e+16

float(738286460302479.0) = 738286460302479.0

float(738286460302479.0) = 738286460302479.0

float(10000000.0) = 10000000.0

float(-0.5) = -0.5

float(9.066612860987508e-243) = 9.066612860987508e-243

float(1.7976931348623157e+308) = 1.7976931348623157e+308

float(1.7976931348623157e+308) = 1.7976931348623157e+308

float(-2.225073858507203e-309) = -2.225073858507203e-309

float(inf) = inf

float(inf) = inf

float(inf) = inf

float(inf) = inf

float(inf) = inf

float(5e-324) = 5e-324

float(-1.1) = -1.1

float(-1.1) = -1.1

float(1e-05) = 1e-05

float(-5.750787780272902e+16) = -5.750787780272902e+16

float(-inf) = -inf

float(-6.103515625e-05) = -6.103515625e-05

float(0.0) = 0.0

float(0.0) = 0.0

float(-2.225073858507203e-309) = -2.225073858507203e-309

float(0.0) = 0.0

float(7.075908598878792e+16) = 7.075908598878792e+16

float(2.8751153895200896e+16) = 2.8751153895200896e+16

float(5e-324) = 5e-324

float(2.220446049250313e-16) = 2.220446049250313e-16

float(-1.827102109746945e+16) = -1.827102109746945e+16

float(-1.827102109746945e+16) = -1.827102109746945e+16

float(-4.340297249901799e+16) = -4.340297249901799e+16

float(-1.5885252397727178e+126) = -1.5885252397727178e+126

float(-1.5885252397727178e+126) = -1.5885252397727178e+126

float(-5.206607255833916e+16) = -5.206607255833916e+16

float(-2.197062345237975e+16) = -2.197062345237975e+16

float(-1.1125369292536007e-308) = -1.1125369292536007e-308

float(0.5) = 0.5

float(-2.225073858507e-311) = -2.225073858507e-311

float(-2.225073858507e-311) = -2.225073858507e-311

float(-2.2250738585072014e-308) = -2.2250738585072014e-308

float(0.5) = 0.5

float(-5e-324) = -5e-324

float(2.00001) = 2.00001

float(1.401298464324817e-45) = 1.401298464324817e-45

float(1.192092896e-07) = 1.192092896e-07

float(5.502504646571906e+16) = 5.502504646571906e+16

float(-inf) = -inf

float(6.103515625e-05) = 6.103515625e-05

float(inf) = inf

float(-1.1) = -1.1

float(-1.1) = -1.1

float(0.99999) = 0.99999

float(0.99999) = 0.99999

float(9007199254740992.0) = 9007199254740992.0

float(1.1458350206854388e+16) = 1.1458350206854388e+16

float(0.99999) = 0.99999

float(-inf) = -inf

float(-10000000.0) = -10000000.0

float(-2.3633787383326325e-132) = -2.3633787383326325e-132

float(6.174447956145954e+16) = 6.174447956145954e+16

float(-2.1157309535032793e-132) = -2.1157309535032793e-132

float(-3.150725624966981e+16) = -3.150725624966981e+16

float(3.150725624966981e+16) = 3.150725624966981e+16

float(3.000013974718901e+16) = 3.000013974718901e+16

float(2.225073858507e-311) = 2.225073858507e-311

float(inf) = inf

float(-1.7976931348623157e+308) = -1.7976931348623157e+308

float(-5.404319552844595e+16) = -5.404319552844595e+16

float(-1.797693134862265e+308) = -1.797693134862265e+308

float(1.797693134862265e+308) = 1.797693134862265e+308

str('') = ''

str('0') = '0'

str('0') = '0'

str('') = ''

str('0') = '0'

str('0') = '0'

str('0') = '0'

str('Ö') = 'Ö'

str('0') = '0'

str('0') = '0'

str('0') = '0'

str('\U000c9847&ö\U00091b08\x99') = '\U000c9847&ö\U00091b08\x99'

str('öÖ\U000a51e0vdÖ\x8b%\x8d\x9cX\x1c}\x83«A峨å¡') =

'öÖ\U000a51e0vdÖ\x8b%\x8d\x9cX\x1c}\x83«A峨å¡'

str('1VÅõ') = '1VÅõ'

str('1VÅõ') = '1VÅõ'

str('1') = '1'

str('11◡̀') = '11◡̀'

str('1\U00010e01×LN\U000423c6C\U000a2400´Ù\x85Ü\x9b\U000c2072\x9e$$') =

'1\U00010e01×LN\U000423c6C\U000a2400´Ù\x85Ü\x9b\U000c2072\x9e$$'

str('1\U00010e01´LN\U000423c6C\U000a2400´Ù\x85Ü\x9b\U000c2072\x9e$$') =

'1\U00010e01´LN\U000423c6C\U000a2400´Ù\x85Ü\x9b\U000c2072\x9e$$'

str('<') = '<'

str('\U000914e2î\U000768a9') = '\U000914e2î\U000768a9'

str('\xadv×\xad\U00060a35\U0010b9ca_') = '\xadv×\xad\U00060a35\U0010b9ca_'

str('\xadv×\xad\U00060a35\U0010b9ca_') = '\xadv×\xad\U00060a35\U0010b9ca_'

str('\U0004ea43\U00056588') = '\U0004ea43\U00056588'

str('嗟ú') = '嗟ú'

str('叟') = '叟'

str('叟ú') = '叟ú'

str('□=\U00071cb0\x11B') = '□=\U00071cb0\x11B'

str('\U000e6000\U0005dc89U%z\x93\U000e90fcs\U000f9651P') =
'\U000e6000\U0005dc89U%z\x93\U000e90fcs\U000f9651P'
str('\U0010185f·') = '\U0010185f·'
str('L?·') = 'L?·'
str('??·') = '??·'
str('?··') = '?··'
str('···') = '···'
str('\U00101db9\U000ffcf0qq©`\x82W\n\U0007be49æ\U000b674dU5Â') =
'\U00101db9\U000ffcf0qq©`\x82W\n\U0007be49æ\U000b674dU5Â'
str('\U000c3a2aÎ') = '\U000c3a2aÎ'
str('\U000e0f01') = '\U000e0f01'
str('9\x9cb\U000460a0«Ð\U00105e0b·') = '9\x9cb\U000460a0«Ð\U00105e0b·'
str('\U0006256f¢') = '\U0006256f¢'
str('\U000aa4fd»±\U0009e121¶myÑ+') = '\U000aa4fd»±\U0009e121¶myÑ+'
str('umé.i¿') = 'umé.i¿'
str('u') = 'u'
str('uu`') = 'uu`'
str('uu愀v¶@\U0005a826è') = 'uu愀v¶@\U0005a826è'
str('uu`') = 'uu`'
str('uu`') = 'uu`'
str('o:U') = 'o:U'
str('i\x8a7Ê<\x04 \U000b1055r\x18w¯.\x03_5') = 'i\x8a7Ê<\x04
\U000b1055r\x18w¯.\x03_5'
str('\U000720b4T') = '\U000720b4T'
str('\U000720b4眞') = '\U000720b4眞'
str('\U000720b4\U000720b4') = '\U000720b4\U000720b4'
str('\U000720b4\U000720b4\x10\x82>') = '\U000720b4\U000720b4\x10\x82>'

str('\U000720b4\U000720b4\x10□>') = '\U000720b4\U000720b4\x10□>'

str('\U000720b4\U000720b4\x103') = '\U000720b4\U000720b4\x103'

str('\U000720b4\U000720b4\x101') = '\U000720b4\U000720b4\x101'

str('\U000bbcc0J¯') = '\U000bbcc0J¯'

str('\U0009cd30\U00076f88\U000ac470') = '\U0009cd30\U00076f88\U000ac470'

str('\U00076f88\U00076f88\U000ac470') = '\U00076f88\U00076f88\U000ac470'

str('\U0005c1ee,u') = '\U0005c1ee,u'

str('\U0008397d\x02í\U000a7a2a/') = '\U0008397d\x02í\U000a7a2a/'

str('µÎ\U0006c85f½') = 'µÎ\U0006c85f½'

str('\U0006e198!\x1bâ\U000de721\x90V\x1e') =

'\U0006e198!\x1bâ\U000de721\x90V\x1e'

str('s') = 's'

str('0²') = '0²'

str('00\x95\x00') = '00\x95\x00'

str('.P\x03ºÞ') = '.P\x03ºÞ'

str('5P\x03ºÞ') = '5P\x03ºÞ'

str('55\x03ºÞ') = '55\x03ºÞ'

str('5') = '5'

str('a\U00071451ù\x01¹õ¢\U0009f03f') = 'a\U00071451ù\x01¹õ¢\U0009f03f'

str('pßú`\x10\x9b0\x88\U00034e3b»&õ\x8b¢XÍ\U000b2659\xa0\x97\t□\U0005529

7嗷') =

'pßú`\x10\x9b0\x88\U00034e3b»&õ\x8b¢XÍ\U000b2659\xa0\x97\t□\U00055297嗷'

str('pßú`\x10\x9b0\x88\U00034e3b»&õ\x8b¢XÍ\U000b2659\xa0\x97\t□\U0005529

7嗷') =

'pßú`\x10\x9b0\x88\U00034e3b»&õ\x8b¢XÍ\U000b2659\xa0\x97\t□\U00055297嗷'

str('\U000b210dÚ-') = '\U000b210dÚ-'

str('\U000bf633') = '\U000bf633'

str('\U0006021b,W\U0001bd7a\U00103fa2') =
'\U0006021b,W\U0001bd7a\U00103fa2'
str('\U0006021b,W\U0001bd7a\U00103fa2') =
'\U0006021b,W\U0001bd7a\U00103fa2'
str('\U000815ff') = '\U000815ff'
str('\U000815ff\x9a\x81u\U0006b322Ù\x0e^\U000f755b\x90Ì\U00079a5f\U000523e6à|È') =
'\U000815ff\x9a\x81u\U0006b322Ù\x0e^\U000f755b\x90Ì\U00079a5f\U000523e6à|È'
str('\U00053d60\r\U00103ff4H±fë*鰄B§\U00083216óú‡') =
'\U00053d60\r\U00103ff4H±fë*鰄B§\U00083216óú‡'
str('\U00105c84å稃') = '\U00105c84å稃'
str('\xa0') = '\xa0'
str('1\x99M\U000975aa\U00038a71¬M') = '1\x99M\U000975aa\U00038a71¬M'
str('*') = '*'
str('*\U0009e20d\x96¦\U000784e4\U0007ec73A/9Í\U000caff2\x9c\x82Ý\x96Ï/\U000f3f9fNþ\U00109c36~\x04ÂÿúÞ7Y(\U000a21e8') =
'*\U0009e20d\x96¦\U000784e4\U0007ec73A/9Í\U000caff2\x9c\x82Ý\x96Ï/\U000f3f9fNþ\U00109c36~\x04ÂÿúÞ7Y(\U000a21e8'
str('*Y\x96¦\U000784e4\U0007ec73A/9Í\U000caff2\x9c\x82Ý\x96Ï/\U000f3f9fNþ\U00109c36~\x04ÂÿúÞ7Y(\U000a21e8') =
'*Y\x96¦\U000784e4\U0007ec73A/9Í\U000caff2\x9c\x82Ý\x96Ï/\U000f3f9fNþ\U00109c36~\x04ÂÿúÞ7Y(\U000a21e8'
str('J7') = 'J7'
str('|Z\U0009aeeb\x85') = '|Z\U0009aeeb\x85'
str('0Z\U0009aeeb\x85') = '0Z\U0009aeeb\x85'
str('\U0009f242') = '\U0009f242'

str('\U000c8460') = '\U000c8460'

str('苓\U000a907e') = '苓\U000a907e'

str('äÎÌ¤v\U00058efev\x14²Q\x88\U000714bdÞÌ\x87Õ\x8c') =

'äÎÌ¤v\U00058efev\x14²Q\x88\U000714bdÞÌ\x87Õ\x8c'

str('\x98W\x07') = '\x98W\x07'

str('2W\x07') = '2W\x07'

str('Y)\x12榷') = 'Y)\x12榷'

str('\x12)\x12榷') = '\x12)\x12榷'

str('È\x14\x97r\x95\x8b\x08') = 'È\x14\x97r\x95\x8b\x08'

str('4/æ嫿(\U0010311bswJ\x9d«7r\U000461ebßúJ\xad') = '4/æ嫿

(\U0010311bswJ\x9d«7r\U000461ebßúJ\xad'

str('\U00048ed1×Â³[') = '\U00048ed1×Â³['

str('\U00048ed1') = '\U00048ed1'

list([]) = []

list([0]) = [0]

list([]) = []

list([0]) = [0]

list([0]) = [0]

list([0]) = [0]

list([0]) = [0]

list([0]) = [0]

list([0]) = [0]

list([0]) = [0]

list([-18135]) = [-18135]

list([111, 67, 48, 8827, 19154, -23734]) = [111, 67, 48, 8827, 19154, -23734]

list([913597224, -5316, -7816, 21394, -30538]) = [913597224, -5316, -7816, 21394,

-30538]

list([913597224, -30538, -7816, 21394, -30538]) = [913597224, -30538, -7816, 21394, -30538]

list([-1388054809]) = [-1388054809]

list([-7800294762277466567, -9885335134104272850613712677404602806 9, 1559598520]) = [-7800294762277466567, -9885335134104272850613712677404602806 9, 1559598520]

list([-5856, -14614, 12783, -44, 26392, 18465, -109, -81, 3393368504693651636, 20090]) = [-5856, -14614, 12783, -44, 26392, 18465, -109, -81, 3393368504693651636, 20090]

list([-40, -549611472682748269, 1581997132, 21104]) = [-40, -549611472682748269, 1581997132, 21104]

list([-40, -549611472682748269, 1581997132, 21104]) = [-40, -549611472682748269, 1581997132, 21104]

list([-30939, -12717]) = [-30939, -12717]

list([-12717, -12717]) = [-12717, -12717]

list([-121, 30318, 5304, 18764, 24108, -2793709654298226305, -23220, -77, -21669, 1954199761893490576]) = [-121, 30318, 5304, 18764, 24108, -2793709654298226305, -23220, -77, -21669, 1954199761893490576]

list([-23220, 30318, 5304, 18764, 24108, -2793709654298226305, -23220, -77, -21669, 1954199761893490576]) = [-23220, 30318, 5304, 18764, 24108, -2793709654298226305, -23220, -77, -21669, 1954199761893490576]

list([-90, -144245402953187457, -3, 18764, 24108, -2793709654298226305, -23220, -77, -21669, 1954199761893490576]) = [-90, -144245402953187457, -3, 18764, 24108, -2793709654298226305, -23220, -77, -21669, 1954199761893490576]

list([10, 62, -1113208835474954089, -28175]) = [10, 62, -1113208835474954089, -28175]

list([27819, 24273, 6860820930769394171, 84, -76, 20222, -5, -7900, 4096, 172808055366964994585889613859128369 18, 4990990383494381158, 28870, 7936, 1967801044, 10360, -14646, 7436, -7012, -94, 4991]) = [27819, 24273, 6860820930769394171, 84, -76, 20222, -5, -7900, 4096, 172808055366964994585889613859128369 18, 4990990383494381158, 28870, 7936, 1967801044, 10360, -14646, 7436, -7012, -94, 4991]

list([101, 4584, -9728225138023411482632076892502364778 2]) = [101, 4584, -9728225138023411482632076892502364778 2]

list([-25859, 4584, -9728225138023411482632076892502364778 2]) = [-25859, 4584, -9728225138023411482632076892502364778 2]

list([-25859]) = [-25859]

list([-25859, -25859, 102, -9850, 16036]) = [-25859, -25859, 102, -9850, 16036]

list([30778, -37, -5576216991921579745, 118, -25993, -6580170632297371439, -161966798523305005995634416436450137 26]) = [30778, -37, -5576216991921579745, 118, -25993, -6580170632297371439, -161966798523305005995634416436450137 26]

list([-96, 11061, 1568388681, -29834, -19799, 567084655]) = [-96, 11061, 1568388681, -29834, -19799, 567084655]

list([18304]) = [18304]

list([-961837264, -3531167647114572423, 343607226246596965848257508872503435 22, 56, -31, -23235]) = [-961837264, -3531167647114572423, 343607226246596965848257508872503435 22, 56, -31, -23235]

list([-8552578, 14916, -541276938619224217, 47, 37, 25231416, -31, -23235]) = [-8552578, 14916, -541276938619224217, 47, 37, 25231416, -31, -23235]

list([0, -72121657778602119, -339031124793176658058547455470299386 88, 56, -31, -23235]) = [0, -72121657778602119, -

339031124793176658058547455470299386 88, 56, -31, -23235]
list([12260, 16350, 7484, 47, 69, -265043604850808475, -20267, -958]) = [12260, 16350, 7484, 47, 69, -265043604850808475, -20267, -958]
list([-14041, 32744, 14576, 136240045310724210209868207969065551622, -10271, 10108, -1523760858, -26757, -30547, -30, -17, 95, 28, 23901, -5618, 1808]) = [-14041, 32744, 14576, 136240045310724210209868207969065551622, -10271, 10108, -1523760858, -26757, -30547, -30, -17, 95, 28, 23901, -5618, 1808]
list([20515]) = [20515]
list([20515, 20515, -26404, -804, 90, -23792, 64]) = [20515, 20515, -26404, -804, 90, -23792, 64]
list([-1170898756, -29265]) = [-1170898756, -29265]
list([12922]) = [12922]
list([12922, 7238]) = [12922, 7238]
list([12922, 7238]) = [12922, 7238]
list([30190]) = [30190]
list([2070325252, 10, 15643]) = [2070325252, 10, 15643]
list([123, 0]) = [123, 0]
list([-22590, -23456, -24049, 24669112097856723440169232953085182105, -8517]) = [-22590, -23456, -24049, 24669112097856723440169232953085182105, -8517]
list([-11341, 16448, 888077513, -2, -23335, 29022]) = [-11341, 16448, 888077513, -2, -23335, 29022]
list([-25535, -29993, -20830, -98, -57]) = [-25535, -29993, -20830, -98, -57]
list([13310, 10871]) = [13310, 10871]
list([32259, 10871]) = [32259, 10871]
list([-18595, 10721, -9337, -93, 30474, 15]) = [-18595, 10721, -9337, -93, 30474, 15]
list([-18035, 10031, 109870955403635324630824434617145373451, 96, -

1173045311, 6092, 94]) = [-18035, 10031,
109870955403635324630824434617145373451, 96, -1173045311, 6092, 94]
list([-25185984947307736942927631673819235489, 90, -7024, -590, 1577237773,
-29154]) = [-25185984947307736942927631673819235489, 90, -7024, -590,
1577237773, -29154]
list([-25185984947307736942927631673819235489, 90, -7024, -590, 1577237773,
-29154]) = [-25185984947307736942927631673819235489, 90, -7024, -590,
1577237773, -29154]
list([317891822, 8813, -1185, 90, -7024, -590, 1577237773, -29154]) = [317891822,
8813, -1185, 90, -7024, -590, 1577237773, -29154]
list([-44, 86, -8964, 29664, 1528433032, 30575]) = [-44, 86, -8964, 29664,
1528433032, 30575]
list([1010126404, 20214]) = [1010126404, 20214]
list([29569]) = [29569]
list([29569, -4810, 5307515931670977042, -
871969778178675838049932219603691477 7, 5365, 12226, 10392, 1060, -11519,
6246, -2760, -4244775724529026560, -31779, -1812564857, -1361562640]) =
[29569, -4810, 5307515931670977042, -
871969778178675838049932219603691477 7, 5365, 12226, 10392, 1060, -11519,
6246, -2760, -4244775724529026560, -31779, -1812564857, -1361562640]
list([-25]) = [-25]
list([0]) = [0]
list([0, -16414]) = [0, -16414]
list([0, -16414, 35, 26782, 28179]) = [0, -16414, 35, 26782, 28179]
list([-21331, -15216, -11325, 6717]) = [-21331, -15216, -11325, 6717]
list([-72, -30347, -48209542399336221 27, -4192, 210, -1630516022, -18537, -
310817319, -3, 27257, -25362, -5, -6854, 94, 23, 14690, -6158, -11418, -13220]) =

[-72, -30347, -4820954239933622127, -4192, 210, -1630516022, -18537, -310817319, -3, 27257, -25362, -5, -6854, 94, 23, 14690, -6158, -11418, -13220]
list([41, -15, -8431]) = [41, -15, -8431]
list([41, -15, -8431]) = [41, -15, -8431]
list([-1591482866]) = [-1591482866]
list([-1591482866, 7366, -17206, 2771, 11, -11207, 21996, 4912972434400995653, 23]) = [-1591482866, 7366, -17206, 2771, 11, -11207, 21996, 4912972434400995653, 23]
list([631667439288922657615572960426471249 5, -29684, -32378, 10025]) = [631667439288922657615572960426471249 5, -29684, -32378, 10025]
list([-14579, -2031523195, -168083004244779650833216101494979948800, 581896938, 17767, -2188, 17131, 44, -71, -2275, 7437, -24029]) = [-14579, -2031523195, -168083004244779650833216101494979948800, 581896938, 17767, -2188, 17131, 44, -71, -2275, 7437, -24029]
list([-1, -3221, 26712]) = [-1, -3221, 26712]
list([-20527, 21658, -8164295892684833852, 6218, -21976, -91, 164332855301124919604200140568071665348, 24869, -62, -65, 28145, 102, 127, 111, 14168, 19016, 5706]) = [-20527, 21658, -8164295892684833852, 6218, -21976, -91, 164332855301124919604200140568071665348, 24869, -62, -65, 28145, 102, 127, 111, 14168, 19016, 5706]
list([9865, -10940, 20983, 29, -6433466211141065777, 10227, -66, -15360, 1, -25800, -75, -1193829528, 19200, -59]) = [9865, -10940, 20983, 29, -6433466211141065777, 10227, -66, -15360, 1, -25800, -75, -1193829528, 19200, -59]
list([7967, -15325, 26343, 26052, -32536, 29510, 32763, 5646]) = [7967, -15325, 26343, 26052, -32536, 29510, 32763, 5646]
list([436230190, 1626]) = [436230190, 1626]

list([-23026, 26802, 1378911005, 38, -126055216149047236272734895851728420211, -5184, -22]) = [-23026, 26802, 1378911005, 38, -126055216149047236272734895851728420211, -5184, -22]
list([-394116526885217519245465103945916458 72]) = [-394116526885217519245465103945916458 72]
list([-394116526885217519245465103945916458 72, -109]) = [-394116526885217519245465103945916458 72, -109]
list([-394116526885217519245465103945916458 72, -394116526885217519245465103945916458 72]) = [-394116526885217519245465103945916458 72, -394116526885217519245465103945916458 72]
list([5367, -3992, 8738, -6478274010883878951, -24505, -2068319649, -10155, -72]) = [5367, -3992, 8738, -6478274010883878951, -24505, -2068319649, -10155, -72]
list([-20, -261653248, 8738, -6478274010883878951, -24505, -2068319649, -10155, -72]) = [-20, -261653248, 8738, -6478274010883878951, -24505, -2068319649, -10155, -72]
list([10285862846166303710691887186958670768 5, 123, 25414, -32406, -24, 12249, -120, 8386, -11, -24, -126, 86, 6875, -20424, -32, -5350356206258758043]) = [10285862846166303710691887186958670768 5, 123, 25414, -32406, -24, 12249, -120, 8386, -11, -24, -126, 86, 6875, -20424, -32, -5350356206258758043]
list([15260, -27228, -27243, 29240, -114, 7547420237122950898, 13627]) = [15260, -27228, -27243, 29240, -114, 7547420237122950898, 13627]
list([52]) = [52]
list([52, -20769, -11, -1786213271, 16987084652846652208438064929041748 2454, -20802, 1842908571201991092, 38, -126, 22685]) = [52, -20769, -11, -1786213271, 16987084652846652208438064929041748 2454, -20802, 1842908571201991092,

38, -126, 22685]
list([52, -20769, -11, -1786213271, 16987084652846652208438064929041748245 4, -20802, 1842908571201991092, 38, -126, 22685]) = [52, -20769, -11, -1786213271, 16987084652846652208438064929041748245 4, -20802, 1842908571201991092, 38, -126, 22685]
list([-19092]) = [-19092]
list([1588116942, 27097, 21807, 566125112502555729, -7]) = [1588116942, 27097, 21807, 566125112502555729, -7]
list([24086, 30723, -12]) = [24086, 30723, -12]
list([24086, 30723, -12]) = [24086, 30723, -12]
list([24086, 0]) = [24086, 0]
list([0, 0]) = [0, 0]
list([949734063253149255790312354496478218 19, 782385934520638826562456284018370142 52, 170653257, 122, -29388, -14017, 6464, -70, -29365, -28970, -3955, 21290, 8384, -20132, -40]) = [949734063253149255790312354496478218 19, 782385934520638826562456284018370142 52, 170653257, 122, -29388, -14017, 6464, -70, -29365, -28970, -3955, 21290, 8384, -20132, -40]
list([21332, -14316, -116, -756665493, -17671, 32313, -6099, -5293]) = [21332, -14316, -116, -756665493, -17671, 32313, -6099, -5293]
list([21332, -14316, -116, -756665493, -17671, 32313, -6099, -5293]) = [21332, -14316, -116, -756665493, -17671, 32313, -6099, -5293]
list([-828, 6146]) = [-828, 6146]
list([-28159, -83, -22694, -20560, -19367, -8199, 15, 24981, 1341]) = [-28159, -83, -22694, -20560, -19367, -8199, 15, 24981, 1341]
dict({}) = {}
dict({'': 0}) = {'': 0}

dict({'': 0}) = {'': 0}

dict({'': 0}) = {'': 0}

dict({'': 0}) = {'': 0}

dict({'': 0}) = {'': 0}

dict({}) = {}

dict({'0': 0}) = {'0': 0}

dict({'': 0}) = {'': 0}

dict({'0': 0}) = {'0': 0}

dict({'': 0}) = {'': 0}

dict({'Ô黏y': 72, '~·ᦩá\x80p÷\x9f\x89籇V\U000d8376\x8c¿': -4552, '': -8846, '颸': 156025062450228808988111745321451549312, 'Ey¸\x89': 23, 'ú7\U00034426õér': -6, '\x83R\x12\x90lë.\U000cb14bÀ\x02': -476559671, 'Lo¶\U000ed990pÃð': 15103}) = {'Ô黏y': 72, '~·ᦩá\x80p÷\x9f\x89籇V\U000d8376\x8c¿': -4552, '': -8846, '颸': 156025062450228808988111745321451549312, 'Ey¸\x89': 23, 'ú7\U00034426õér': -6, '\x83R\x12\x90lë.\U000cb14bÀ\x02': -476559671, 'Lo¶\U000ed990pÃð': 15103}

dict({'\U000de7c0ë2¥': 19257, 'l': -3384, 'Ñ\x9a²^\x9f\x90®e\x01\x8c᪥~Ü\x9e\U0004862cî\x18': 109}) = {'\U000de7c0ë2¥': 19257, 'l': -3384, 'Ñ\x9a²^\x9f\x90®e\x01\x8c᪥~Ü\x9e\U0004862cî\x18': 109}

dict({'¦\U0001425cÒú\U00059f43': -319983933}) = {'¦\U0001425cÒú\U00059f43': -319983933}

dict({'Ý\x80\x93': 29761, '': -25599, 'P\x8có': -6306}) = {'Ý\x80\x93': 29761, '': -25599, 'P\x8có': -6306}

dict({'ÅÇ': 6691, '': -2772, ' D芷*\x9f£I18ñ\x96': -9089392529299193313633813373729418230 6, '=\x05\x13N\\ä\x1b-\x8c\x9d\x87l'\x03{\U0001dfe7{'': -9769, '\U000a490c': 20434, '\x17': 4743}) =

{'ÅÇ': 6691, '': -2772, ' D芷*\x9f£I18ñ\x96': -
90893925292991933136338133737294182306, '=\x05\x13N\\ä\x1b-
\x8c\x9d\x87l'\x03{\U0001dfe7{'': -9769, '\U000a490c': 20434, '\x17': 4743}
dict({'\r': -29633, '\U000d6b47\U0007802c\x8bÜ': 14068, '\x10\x9fÀ': 27224,
'?vÜq7óËw': -5362477828559235398, '': 7834}) = {'\r': -29633,
'\U000d6b47\U0007802c\x8bÜ': 14068, '\x10\x9fÀ': 27224, '?vÜq7óËw': -
5362477828559235398, '': 7834}
dict({'ü+\U000936bat\xa0\U000f5e14ᾀ': -16328, 'à\r□': 32613, '': -18707,
'%\x9aV\x1bZ\U000944e0\x84\x92#lÊ®ÐY': 8, 'Þ\
U00090123\x95Îg-': -12, '«': 582, '\U000eff71\x0cê(«¼\U0005364f': -42,
'¿]\U000ca844g\x19\U0006670c`e': -13301, '¨\x97': 8,
'Õ¯\x15g|^Á\U0005348e4\x04\U0007ddb3Ü=\x8c\x1d\U000d0170B\x87Y\U000f1
883': -20665}) = {'ü+\U000936bat\xa0\U000f5e14�': -16328, 'à\r□': 32613, '': -
18707, '%\x9aV\x1bZ\U000944e0\x84\x92#lÊ®ÐY': 8, 'Þ\U00090123\x95Îg-': -12,
'«': 582, '\U000eff71\x0cê(«¼\U0005364f': -42, '¿]\U000ca844g\x19\U0006670c`e':
-13301, '¨\x97': 8,
'Õ¯\x15g|^Á\U0005348e4\x04\U0007ddb3Ü=\x8c\x1d\U000d0170B\x87Y\U000f1
883': -20665}
dict({'ªsFÝ\U00096eb0ð¶#\x0c': 21337, '¡\U0003a86c': -1, '\U0008e206B\\醞': -
4114, '\x9b\x0e·\x8d': -127, '\xa0\x18î\U000e2062\U000d0c07à9簽Í\U000960db': -
30, '': -7013644158805338313849413304854915211 2, '\U0007d3d9´@\x0cA': -
278458770, '\U000c780aà¢¿': 73,
'a\U0004d5f9T.p\x89N\U000bfaafÿ\U000b45c4ï\r': 90, '\x1aÉ6kA6d< \x94O.': 85,
'\U00044dc1q\x94Bü\x16ÜíÛ"§[\U0006f5bd¤\U000a3386': -
3566941456489509576, '\x1a\U00075fcc>:Y\x0c': 1038,
'□o¬\U000f37adQ\U00081e165\x0cF\x95¾\x80': 12727, '\\': -30103, '2': -56}) =

{'ªsFÝ\U00096eb0ð¶#\x0c': 21337, '¡\U0003a86c': -1, '\U0008e206B\\醞': -4114,
'\x9b\x0e·\x8d': -127, '\xa0\x18î\U000e2062\U000d0c07à9箋Í\U000960db': -30, '': -
70136441588053383138494133048549152112, '\U0007d3d9´@\x0cA': -278458770,
'\U000c780aà¢¿': 73, 'a\U0004d5f9T.p\x89N\U000bfaafÿ\U000b45c4ï\r': 90,
'\x1aÉ6kA6d< \x94O.': 85,
'\U00044dc1q\x94Bü\x16ÜíÛ"§[\U0006f5bd¤\U000a3386': -
3566941456489509576, '\x1a\U00075fcc>:Y\x0c': 1038,
'□o¬\U000f37adQ\U00081e165\x0cF\x95¾\x80': 12727, '\\': -30103, '2': -56}
dict({'í\x9cÜ\x0b\x92\U00074afevÎ': 829768096,
'çaá\x9dx\x9e\x0f\U000bb1b0\U0005ec046~ÓÒ\x1c\x1dà\x943̃\x0e': 76,
'\U00059975': -20173, 'ñç\x11\x8e\U0001a7b9¹Õ': -31364}) =
{'í\x9cÜ\x0b\x92\U00074afevÎ': 829768096,
'çaá\x9dx\x9e\x0f\U000bb1b0\U0005ec046~ÓÒ\x1c\x1dà\x943̃\x0e': 76,
'\U00059975': -20173, 'ñç\x11\x8e\U0001a7b9¹Õ': -31364}
dict({'': -80, ')Å\x81': 7041, '\x14ØÔ': -95, 'Y@': -119}) = {'': -80, ')Å\x81': 7041,
'\x14ØÔ': -95, 'Y@': -119}
dict({'': 7041, ')Å\x81': 7041, '\x14ØÔ': -95, 'Y@': -119}) = {'': 7041, ')Å\x81': 7041,
'\x14ØÔ': -95, 'Y@': -119}
dict({'': -27, '2': 50356867, 'ٰ': -3, '12Ô': -95, 'Y@': -119}) = {'': -27, '2': 50356867, 'ٰ':
-3, '12Ô': -95, 'Y@': -119}
dict({'#': 21287, 'ç\x99\x0e¯': 110, 'J\U0006304b\U000d0c6eÂÇOt7,[Ö\t\x1b\x12': -
16964}) = {'#': 21287, 'ç\x99\x0e¯': 110,
'J\U0006304b\U000d0c6eÂÇOt7,[Ö\t\x1b\x12': -16964}
dict({'': -30107, '\U000a1956': -82, 'Ý\x98;\x0b': -3209, '\x05': -18237}) = {'': -
30107, '\U000a1956': -82, 'Ý\x98;\x0b': -3209, '\x05': -18237}
dict({'': -72}) = {'': -72}
dict({'': -16, '\x0b': -4211298107305119547, 'IY': 4222, '遊

\U000e2bc2d\U000e4447©3+£\x7f1C': 2226566938762648114, '\U000386b2Á': -59, '\U00047350·': 1134214373, '\x94\x06\x06\x8c}\x89¥Ø\x87\x8bð\U00083eb2\x15`&\x87': -174809043, '\x90\x8bxó\x8f\x1aÈ': 775496500, '\U0009dd3fÂÃºÖñï': 36}) = {'': -16, '\x0b': -4211298107305119547, 'IY': 4222, '遊\U000e2bc2d\U000e4447©3+£\x7f1C': 2226566938762648114, '\U000386b2Á': -59, '\U00047350·': 1134214373, '\x94\x06\x06\x8c}\x89¥Ø\x87\x8bð\U00083eb2\x15`&\x87': -174809043, '\x90\x8bxó\x8f\x1aÈ': 775496500, '\U0009dd3fÂÃºÖñï': 36}

dict({'ʯÆ\x83$\x08': -6, '\x80X\U000e0aa1\xad\x95': -6, '\U000379dbám\x1eo\x9a\U000da8fd\x8cS\x1f\U000f65bf': 30612, '\U000f7d82\\p\x94ûqoÉ': -5080547204592589107, '\U0004b19b\x04\U000906beë\x1e\nÀË7%\x19\x83\x95ù\U000d812e': -3670905972559888318}) = {'ʯÆ\x83$\x08': -6, '\x80X\U000e0aa1\xad\x95': -6, '\U000379dbám\x1eo\x9a\U000da8fd\x8cS\x1f\U000f65bf': 30612, '\U000f7d82\\p\x94ûqoÉ': -5080547204592589107, '\U0004b19b\x04\U000906beë\x1e\nÀË7%\x19\x83\x95ù\U000d812e': -3670905972559888318}

dict({'(\x01\U00109c80\x06': -4038068910237755992481033866739911526}) = {'(\x01\U00109c80\x06': -4038068910237755992481033866739911526}

dict({'': -25269, '\U0010b52f・': 57, 'Zìñ¯\x8c``äþËå®ä\x9cLÛ': -1905, '\x83Yô\U0001900bÖzb': 46, 'ù3\U0004227e\x04': 25524, ' \x9e': 11819, '\U000d1847QÂî\U00105a6f\x85\x96_\x9fa': 29223, '\U0006add0¨û': 157020045507278338084941763592887810608, 'L¤': -28717, '\U000e293e_x\U0005f6d6\U0001174c\U0009073f': -127, '\U000ccb94S?Æ\U000e3472\x95»ï9ÉµD\U0005b800□\U000f8f7f\x007N': 23, '\xad\x9c(éO': -13741}) = {'': -25269, '\U0010b52f・': 57,

'Zìñ¯\x8c``äþËå®ä\x9cLÛ': -1905, '\x83Yô\U0001900bÖzb': 46,
'ù3\U0004227e\x04': 25524, ' \x9e': 11819,
'\U000d1847QÂî\U00105a6f\x85\x96_\x9fa': 29223, '\U0006add0¨û':
157020045507278338084941763592887810608, 'L¤': -28717,
'\U000e293e_x\U0005f6d6\U0001174c\U0009073f': -127,
'\U000ccb94S?Æ\U000e3472\x95»ï9ÉµD\U0005b800□\U000f8f7f\x007N': 23,
'\xad\x9c(éO': -13741}
dict({'2\U0010ddae}þ\x8cÞ\U0008474e\x1d': 21518,
'\x17C\x8d\U000fa995>\x9e¼k\x05\x8a': -10104, ' h¨ä\x9a\U000a5a7d': -2784, '':
27079, 'Ç+\U00064e68\x8e\U000cb751': -16615,
'¢\x9f¯yÝQ\x0f\x05w9\x12C\x0fi¨\x1c\x80\n\x89\xa0µDZ□Ñ\U000dc29f?èù|□□':
7394, 'º\x9d': -9252, '\x9dO\U00106b24\x9bSw': 18701}) =
{'2\U0010ddae}þ\x8cÞ\U0008474e\x1d': 21518,
'\x17C\x8d\U000fa995>\x9e¼k\x05\x8a': -10104, ' h¨ä\x9a\U000a5a7d': -2784, '':
27079, 'Ç+\U00064e68\x8e\U000cb751': -16615,
'¢\x9f¯yÝQ\x0f\x05w9\x12C\x0fi¨\x1c\x80\n\x89\xa0µDZ□Ñ\U000dc29f?èù|□□':
7394, 'º\x9d': -9252, '\x9dO\U00106b24\x9bSw': 18701}
dict({'': -25621, 'kg': -14237, '\x04á\x19': 14, '\x94': -28,
'\r¨xì\x7f\x94¡\U000a035a~ù\x9fÒ-\x11ú\U000c40a1'·M,\U0003febd=': 80}) = {'':
-25621, 'kg': -14237, '\x04á\x19': 14, '\x94': -28, '\r¨xì\x7f\x94¡\U000a035a~ù\x9fÒ-
\x11ú\U000c40a1'·M,\U0003febd=': 80}
dict({'': 5598899245239852967, '\U000e1d7f': -1834707347,
'AT\x91®¨À\xa0\x1fñ(¬F^': 145441596379274862256996946826341937212,
'ý1+\x9ebß\U00105023\U000c90ad\x84': 20472,
'\U00065888□\x89ú\x86!1\U000a6e2d\U00093225[': 31, '-': -14, 'zh': 6290,
'M(ý\U000a7698': 15, 'ê': 40, 'i': -18142}) = {'': 5598899245239852967,
'\U000e1d7f': -1834707347, 'AT\x91®¨À\xa0\x1fñ(¬F^':

145441596379274862256996946826341937212,
'ý1+\x9ebß\U00105023\U000c90ad\x84': 20472,
'\U00065888□\x89ú\x86!1\U000a6e2d\U00093225[': 31, '-': -14, 'zh': 6290,
'M(ý\U000a7698': 15, 'ê': 40, 'i': -18142}
dict({'': -62, 'ä4': 71, '¿¾\U0005224e': 30678,
'\U000af570\x97\x8bG\x0e\U0003fe93ª': 1275, '[x': -25330, '\x99':
154959473394931837695635863628233174872,
'|\x9a¼r\U000d71cc²\x16È\U0010d137j\U0006ae20D': -20837,
'ßñÐ\U000c15a2_\x85\U000fb263ù\U000b9094ò\x8aÎ\U000c6f48': -20421,
'ÝSþÍSwÛnïO\x01\U00079e00¹ÒþÚ\x07i\x9eÙ¸Yð\x8e\U000d9be6w#<O\x8f纗': -
42, '\U000c40af¨\x16\U000ee868': 54, '\U000f64b4ÁµWV': -20794, '\x00Ã\x85G': -
4648, '±': -107}) = {'': -62, 'ä4': 71, '¿¾\U0005224e': 30678,
'\U000af570\x97\x8bG\x0e\U0003fe93ª': 1275, '[x': -25330, '\x99':
154959473394931837695635863628233174872,
'|\x9a¼r\U000d71cc²\x16È\U0010d137j\U0006ae20D': -20837,
'ßñÐ\U000c15a2_\x85\U000fb263ù\U000b9094ò\x8aÎ\U000c6f48': -20421,
'ÝSþÍSwÛnïO\x01\U00079e00¹ÒþÚ\x07i\x9eÙ¸Yð\x8e\U000d9be6w#<O\x8f纗': -
42, '\U000c40af¨\x16\U000ee868': 54, '\U000f64b4ÁµWV': -20794, '\x00Ã\x85G': -
4648, '±': -107}
dict({'': 7661, '¾{': -21741, '®\U00044591½û\x88\U000c3b92ØÎü': 5871, '\x8a': -87,
'¨Ë¾\x9còµ\x9b\U00068702ÌØ\x0e{È8\x7f£9ò\x1d\U000aff99ãø6û\x9e\U0009d160
\x96\U000ffb0dü\x0cy)¸Î¯': 32654, '';.\x9e'": -24219, '\x8e': -17971, "\x05vTB'":
32738, '©J\x04': 3567, '\x04§\x1e\r¾': -89, '\x17½': -
10341443499037245781209948503116925182, '\U00072266\xad': 27593}) = {'':
7661, '¾{': -21741, '®\U00044591½û\x88\U000c3b92ØÎü': 5871, '\x8a': -87,
'¨Ë¾\x9còµ\x9b\U00068702ÌØ\x0e{È8\x7f£9ò\x1d\U000aff99ãø6û\x9e\U0009d160
\x96\U000ffb0dü\x0cy)¸Î¯': 32654, '';.\x9e'": -24219, '\x8e': -17971, "\x05vTB'":

32738, '©J\x04': 3567, '\x04§\x1e\r¾': -89, '\x17½': -
10341443499037245781209948503116925182, '\U00072266\xad': 27593}
dict({'': -8581}) = {'': -8581}
dict({'*z': 1027276677, '&É7(\x02+\U0009c1f0\x10W\U0007c828\x06': -
10745365168633714373979484235435854975 4, '': 11882, '\x1fp§\U00075f3a5': -
20027}) = {'*z': 1027276677, '&É7(\x02+\U0009c1f0\x10W\U0007c828\x06': -
10745365168633714373979484235435854975 4, '': 11882, '\x1fp§\U00075f3a5': -
20027}
dict({'': 1027276677, '&É7(\x02+\U0009c1f0\x10W\U0007c828\x06': -
10745365168633714373979484235435854975 4, '\\\U00040a05': -
216234359068491859, '\x8aD13255': 2□, '\U0007cca0-Q\U000ac0a6': -86, 'zÿ7□': -
3363818168104774811, 'ý¡f\\Ö\x11z¾J\U
00068aec': -619655916069018356776377431810173439 06, '\U00032527\x00': 14,
'\U000d02fc': 32259, 'gá\x8b\U0006d0a1û\x96': -20598,
'"\U000bef0e\U000164ffÝ'\xa0s#ÅÒ": -11376, '\x83ú3d\U0002ed70\U00040831¸':
20282}) = {'': 1027276677, '&É7(\x02+\U0009c1f0\x10W\U0007c828\x06': -
10745365168633714373979484235435854975 4, '\\\U00040a05': -
216234359068491859, '\x8aD2□'
: 13255, '\U0007cca0-Q\U000ac0a6': -86, 'zÿ7□': -3363818168104774811,
'ý¡f\\Ö\x11z¾J\U00068aec': -619655916069018356776377431810173439 06,
'\U00032527\x00': 14, '\U000d02fc': 32259, 'gá\x8b\U0006d0a1û\x96': -20598,
'"\U000bef0e\U000164ffÝ'\xa0s#ÅÒ": -11376, '\x83ú3d\U0002ed70\U00040831¸':
20282}
dict({'': 21286, '*\U000f3002\U000c428a\U0007d190íµ\U000dd735\x14ï\x0c': -
10379, '\x0f\U00108c35W\U000d3ae9w\x03': 25114,
'HVµÇ\U00063d19\U000f96c0Ã\U000b78b1\U000eb40fÑ\U00081444\U0003952b
O': 8714, 'ýÞ': -9361, 'u¶ª\x89/': -403829059, '\x9e\U0005bb6b´Ã\x8dM': -13499,

"Î\x83"': -7948511791579275002880883729094992021 8, 'D\x1b': 11804932}) = {": 21286, '*\U000f3002\U000c428a\U0007d190íµ\U000dd735\x14ï\x0c': -10379, '\x0f\U00108c35W\U000d3ae9w\x03': 25114, 'HVµÇ\U00063d19\U000f96c0Ã\U000b78b1\U000eb40fÑ\U00081444\U0003952b O': 8714, 'ýÞ': -9361, 'u¶ª\x89/': -403829059, '\x9e\U0005bb6b´Ã\x8dM': -13499, "Î\x83"': -7948511791579275002880883729094992021 8, 'D\x1b': 11804932}

dict({": 7826, '\U0008066e\x9e\U00092af4µì0': 19856, '\U0003aa11ì¼': 2672114804754954636, '\x1eá□2□H×': -27, 'º\U0007c014': -21192}) = {": 7826, '\U0008066e\x9e\U00092af4µì0': 19856, '\U0003aa11ì¼': 2672114804754954636, '\x1eá□2□H×': -27, 'º\U0007c014': -21192}

dict({": -27, '\U0008066e\x9e\U00092af4µì0': 19856, '\U0003aa11ì¼': 2672114804754954636, '\x1eá□2□H×': -27, 'º\U0007c014': -21192}) = {": -27, '\U0008066e\x9e\U00092af4µì0': 19856, '\U0003aa11ì¼': 2672114804754954636, '\x1eá□2□H×': -27, 'º\U0007c014': -21192}

dict({": -27, '\U0008066e\x9e\U00092af4µì0': 19856, '\U0003aa11ì¼': 2672114804754954636, '\x1eá□2□H×': -27, 'º\U0007c014': -21192}) = {": -27, '\U0008066e\x9e\U00092af4µì0': 19856, '\U0003aa11ì¼': 2672114804754954636, '\x1eá□2□H×': -27, 'º\U0007c014': -21192}

dict({'®û$': -35, '\U000c36e0': -60}) = {'®û$': -35, '\U000c36e0': -60}

dict({": 29502, '5': -94, "'¸\x86\x82": 511292670}) = {": 29502, '5': -94, "'¸\x86\x82": 511292670}

dict({": 29502, "'¸\x86\x82": 511292670, "1\U000d525a\U00048b7aítc'\t\x17Á": 105, '²\U00048a49Ù\U000f0b18\x04D': 52, '+¤ýÉµz\U000b8ccc\xad\U000b61ad\x17\x04': -1251524107}) = {": 29502, "'¸\x86\x82": 511292670, "1\U000d525a\U00048b7aítc'\t\x17Á": 105, '²\U00048a49Ù\U000f0b18\x04D': 52, '+¤ýÉµz\U000b8ccc\xad\U000b61ad\x17\x04': -1251524107}

dict({'«¥¡ô': 30014}) = {'«¥¡ô': 30014}

dict({'': 1842968068, 'T\x03\x1e+灶8¨Â濬': 22051, 'c\x187\U0009c97a': 13, '\x90t': -22028, 'Å!\x93»\x1aµ': -25630, '\x19\x8dÁ)M': 838928901973568555544819328669879418 29, '\x94': -163277735500996320751896236361027521109, 'ðn\U00055f12ÿ^L\x90': 17932}) = {'': 1842968068, 'T\x03\x1e+灶8¨Â濬': 22051, 'c\x187\U0009c97a': 13, '\x90t': -22028, 'Å!\x93»\x1aµ': -25630, '\x19\x8dÁ)M': 838928901973568555544819328669879418 29, '\x94': -163277735500996320751896236361027521109, 'ðn\U00055f12ÿ^L\x90': 17932}

dict({'': 28121, '5': 512, '11H2': -262784, 'Â濬': 22051, 'c\x187\U0009c97a': 13, '\x90t': -22028, 'Å!\x93»\x1aµ': -25630, '\x19\x8dÁ)M': 838928901973568555544819328669879418 29, '\x94': -163277735500996320751896236361027521109, 'ðn\U00055f12ÿ^L\x90': 17932}) = {'': 28121, '5': 512, '11H2': -262784, 'Â濬': 22051, 'c\x187\U0009c97a': 13, '\x90t': -22028, 'Å!\x93»\x1aµ': -25630, '\x19\x8dÁ)M': 838928901973568555544819328669879418 29, '\x94': -163277735500996320751896236361027521109, 'ðn\U00055f12ÿ^L\x90': 17932}

dict({'': -30880, '\x9e': -38, '\U000f4e64¥[\x8b': -28375, '%á': -94, 'Õ`+Ü\U0007ed47': 35}) = {'': -30880, '\x9e': -38, '\U000f4e64¥[\x8b': -28375, '%á': -94, 'Õ`+Ü\U0007ed47': 35}

dict({'\U000c5bba': 120}) = {'\U000c5bba': 120}

dict({'0': 0}) = {'0': 0}

dict({'': 0}) = {'': 0}

dict({'': -2}) = {'': -2}

dict({'煨\x91"\U00101c608\x83-É¢ ¾Ü"\x96:Lm': 10787, 'X\\\x98\x80b\x8c\U000815b75\U0007cbf4': -1256, 'P\x95A\U000ecfdf': 32296}) =

{'煨\x91"\U00101c608\x83-É¢ ¾Ü"\x96:Lm': 10787, 'X\\\x98\x80b\x8c\U000815b75\U0007cbf4': -1256, 'P\x95A\U000ecfdf': 32296}
dict({'': 84, '\x85': -98, 'Æ': -6087991671556483922, '\x84ß{È\x8dMÆ': -26177, '|睿': 51, 'shK¥êÏ\x01\U000ae272': 26251}) = {'': 84, '\x85': -98, 'Æ': -6087991671556483922, '\x84ß{È\x8dMÆ': -26177, '|睿': 51, 'shK¥êÏ\x01\U000ae272': 26251}
dict({'\U000d1786\x80': 5436, '\x87\x0c': -118, 'Í\U000c6b0e*&\xad\x1bi\x04Ç«"縿ð\U00106924e': 19, '': 26488, 'o\x14\U000cb7556\U0009be00P': -9651, 'Á': 25270, 'c': -64, '\xadiËgo\U0001bd0dÂ': -21394, 'z': -35, '}½\x8c': -7980, ']\x9d': 9378, 'Ü': 22155, '\U000841a3\x11;Sg': -67, 'S': -17924, 'óg\x9e': 7426, '0\U000aa39b': 15, 'ª': 122, "'¤\x07_4\U000c9d49&": -91}) = {'\U000d1786\x80': 5436, '\x87\x0c': -118, 'Í\U000c6b0e*&\xad\x1bi\x04Ç«"縿ð\U00106924e': 19, '': 26488, 'o\x14\U000cb7556\U0009be00P': -9651, 'Á': 25270, 'c': -64, '\xadiËgo\U0001bd0dÂ': -21394, 'z': -35, '}½\x8c': -7980, ']\x9d': 9378, 'Ü': 22155, '\U000841a3\x11;Sg': -67, 'S': -17924, 'óg\x9e': 7426, '0\U000aa39b': 15, 'ª': 122, "'¤\x07_4\U000c9d49&": -91}
dict({'': -31339}) = {'': -31339}
dict({'': -17678, '\x91÷': 39, '\x8d\x00×\U000e30bd\U0006ceb3®l\U0007fe731t': 2277732941042499941, '\x94': 28915}) = {'': -17678, '\x91÷': 39, '\x8d\x00×\U000e30bd\U0006ceb3®l\U0007fe731t': 2277732941042499941, '\x94': 28915}
dict({'': -17678, '\x91÷': 39, '\x8d\x00×\U000e30bd\U0006ceb3®l\U0007fe731t': 2277732941042499941, '\x94': 28915}) = {'': -17678, '\x91÷': 39, '\x8d\x00×\U000e30bd\U0006ceb3®l\U0007fe731t': 2277732941042499941, '\x94': 28915}
dict({'\U000b835fP¿': 586, '%Ý\n\U0010f498¤□\x00\x92í': 1501661585, '\x1d\x04í\U000d4001': 24264}) = {'\U000b835fP¿': 586,

'%Ý\n\U0010f498¤□\x00\x92í': 1501661585, '\x1d\x04í\U000d4001': 24264}
dict({'': 27923, 'àÁÞ\x0f¢\U000b8d95ãÙ#òñ5': 23812,
'¥Ò`\x86\x12m\U000efd8a\x98鰯': -100, 'u\U00053d4fx\xadÏy
Î\U000d4811OÖOÎ¶□□': -20900}) = {'': 27923, 'àÁÞ\x0f¢\U000b8d95ãÙ#òñ5':
23812, '¥Ò`\x86\x12m\U000efd8a\x98鰯': -100, 'u\U00053d4
fx\xadÏy Î\U000d4811OÖOÎ¶□□': -20900}
dict({'Þ£\U000384d5V\x1e¶üB\U000991bd\xa0+X': 1084110024, '': 2033603622,
'ô\x0b\U0006a563k\x9c\r\U00107f40@\U000deb5e': -2925}) =
{'Þ£\U000384d5V\x1e¶üB\U000991bd\xa0+X': 1084110024, '': 2033603622,
'ô\x0b\U0006a563k\x9c\r\U00107f40@\U000deb5e': -2925}
dict({'': -15, '\x06à-': 18647, 'B\U0005e6f2': 1138, 'h': 17612,
'ßZ.ÒÇ\x08O:º\x9b\x01+L\U00104ebaª\U000e7641¡': 86, 's\x82\x9b': -49, 'ö': -1705,
'\x96e큊\U0004a4c0\x1a\x8c': 1468491867230085814, 'Ò½\xa0\xa0\x9b\r': -
328762784, '\U000fd5ac¥4': 124}) = {'': -15, '\x06à-': 18647, 'B\U0005e6f2': 1138,
'h': 17612, 'ßZ.ÒÇ\x08O:º\x9b\x01+L\U00104ebaª\U000e7641¡': 86, 's\x82\x9b': -49,
'ö': -1705, '\x96e큊\U0004a4c0\x1a\x8c': 1468491867230085814,
'Ò½\xa0\xa0\x9b\r': -328762784, '\U000fd5ac¥4': 124}
dict({'': 1008, 'c\U00044679Ú\U0006ff17\x8a\U000dd3f2\x8at\U00079345éÌÔ':
8339, 'ª\U000f8f8a\x9c\U00083663\x0b\x13\x13\U0003b2b9#]É/yÇz;ë[': 7993,
'\x82': -51, '\U000adf70\x04\x10': 10814}) = {'': 1008,
'c\U00044679Ú\U0006ff17\x8a\U000dd3f2\x8at\U00079345éÌÔ': 8339,
'ª\U000f8f8a\x9c\U00083663\x0b\x13\x13\U0003b2b9#]É/yÇz;ë[': 7993, '\x82': -51,
'\U000adf70\x04\x10': 10814}
dict({'': -51, 'c\U00044679Ú\U0006ff17\x8a\U000dd3f2\x8at\U00079345éÌÔ': 8339,
'ª\U000f8f8a\x9c\U00083663\x0b\x13\x13\U0003b2b9#]É/yÇz;ë[': 7993, '\x82': -51,
'\U000adf70\x04\x10': 10814}) = {'': -51,
'c\U00044679Ú\U0006ff17\x8a\U000dd3f2\x8at\U00079345éÌÔ': 8339,

'ª\U000f8f8a\x9c\U00083663\x0b\x13\x13\U0003b2b9#]É/yÇz;ë[': 7993, '\x82': -51, '\U000adf70\x04\x10': 10814}

dict({'': -8409567855690299216, '\x8då': -114}) = {'': -8409567855690299216, '\x8då': -114}

dict({'': -1958005096, '勦\x8då': -114}) = {'': -1958005096, '勦\x8då': -114}

dict({'': -1958005096}) = {'': -1958005096}

dict({'': -116, 'À\U0004db76Í耔': -9984, '\U0004a58fø': -18404, '¢\x19\x08\rB\U0002ecdd\U00059366U\x16°': 11037}) = {'': -116, 'À\U0004db76Í耔': -9984, '\U0004a58fø': -18404, '¢\x19\x08\rB\U0002ecdd\U00059366U\x16°': 11037}

dict({'嵼': -1064114735, '\U0006b6e5\x94\U0009f5c8': 1283}) = {'嵼': -1064114735, '\U0006b6e5\x94\U0009f5c8': 1283}

dict({'': 26946, 'y\U0006df46\x1c': -28, '°\x8dO\x07G\U000b3b18': 26473, '\x1ah¯jÔÈ;': -25001}) = {'': 26946, 'y\U0006df46\x1c': -28, '°\x8dO\x07G\U000b3b18': 26473, '\x1ah¯jÔÈ;': -25001}

dict({'¸\U000992d1': 6800806494596113391283900050161459059 6}) = {'¸\U000992d1': 6800806494596113391283900050161459059 6}

dict({'¸\U000992d1': 6800806494596113391283900050161459059 6}) = {'¸\U000992d1': 6800806494596113391283900050161459059 6}

dict({'\U000992d1': 6800806494596113391283900050161459059 6}) = {'\U000992d1': 6800806494596113391283900050161459059 6}

dict({'\U000bfaef': 83, 'h\U0005c71e\x07\x94#¡n¾¡\x82\x07\x9cÕ\U000bde34ÆaëÀ\U0009b891`ù-\x1c\x81': -1874333508}) = {'\U000bfaef': 83, 'h\U0005c71e\x07\x94#¡n¾¡\x82\x07\x9cÕ\U000bde34ÆaëÀ\U0009b891`ù-\x1c\x81': -1874333508}

dict({'\U000f0380': 50, 'ràà': 42, 'D_': -29128, '³ßùz¸': -4227864030024707602,

'\U00037c32': -1146767143,
'\x0e\U0003ce78Ø¨\U0005f8b4r\U0006f0d4\U000d6726$t': -2440, '\U000c9a17n\\':
22871, 'àüÅd\U0006121a\U0008cadb': -26278, '\U0004d18f\x99\U0001a0b1': -
31388, '°': -804188183}) = {'\U000f0380': 50, 'ràà': 42, 'D_': -29128, '³ßùz¸': -
4227864030024707602, '\U00037c32': -1146767143,
'\x0e\U0003ce78Ø¨\U0005f8b4r\U0006f0d4\U000d6726$t': -2440, '\U000c9a17n\\':
22871, 'àüÅd\U0006121a\U0008cadb': -26278, '\U0004d18f\x99\U0001a0b1': -
31388, '°': -804188183}
dict({'h': -8006, 'Ì\x11àF*\x90': 7046, 'Áî^\U00085924þ\x02GZ\U0001c72d&)iY': -
3231, 'ÓOó\U0003d42b': 79, 'A': 20380,
'\U000945f9jTy{\U00049e7b\x0b\U000ccb8a': 8296, '': -28542, 'ì\\\x1b':
1751547820, '{': -15, '\x19T\x7f\U000d543c:\U000bab79¢翌\U000c04b7': 91,
'\U0007b3f3\x93¡\x10Wæ\x02®BmNÜ\x9e': -68334635, '9ì\x99': -
72383849569917814197824458667602085793, 'f_': -75, '6蔑': 87}) = {'h': -8006,
'Ì\x11àF*\x90': 7046, 'Áî^\U00085924þ\x02GZ\U0001c72d&)iY': -3231,
'ÓOó\U0003d42b': 79, 'A': 20380, '\U000945f9jTy{\U00049e7b\x0b\U000ccb8a':
8296, '': -28542, 'ì\\\x1b': 1751547820, '{': -15,
'\x19T\x7f\U000d543c:\U000bab79¢翌\U000c04b7': 91,
'\U0007b3f3\x93¡\x10Wæ\x02®BmNÜ\x9e': -68334635, '9ì\x99': -
72383849569917814197824458667602085793, 'f_': -75, '6蔑': 87}
dict({'': -1289178561488982075, 'lwÝ£ \x07\x7f': -516, '\x9a\U000840ff': 4770,
'\x16»': -1041916607, '\x17\U000e4e94RN7ÿ\x9f1蕉
öxU\U000993d3\U0003b520\x90': 34, '{ö': -18, '²\U000cd141\U0001e530': 4, 'ù': 15,
'Â:\U000d8e59í': -1983472690, '³\x02\x00²·\x83\U000191c4\x81\U000ee95e
6\U000dcb75Ý': -29857, '¬': 10979, '[\U00079b98\nÞ}F': 28,
'"\U000d5b6d\x86b¨\U000f840d3\U000d1639'î2ÿ\x9f\U0007a120'': -

1724151301267494345, '\U0009167aØ': -20643, '\U000a2c62': 4531811023802960318, '\U00093676Ö«\x1d¡\x85': -922}) = {'': -1289178561488982075, 'lwÝ£ \x07\x7f': -516, '\x9a\U000840ff': 4770, '\x16»': -1041916607, '\x17\U000e4e94RN7ÿ\x9f1蒸öxU\U000993d3\U0003b520\x90': 34, '{ö': -18, '²\U000cd141\U0001e530': 4, 'ù': 15, 'Â:\U000d8e59í': -1983472690, '³\x02\x00²·\x83\U000191c4\x81\U000ee95e 6\U000dcb75Ý': -29857, '¬': 10979, '[\U00079b98\nÞ}F': 28, "\U000d5b6d\x86b¨\U000f840d3\U000d1639î2ÿ\x9f\U0007a120": -1724151301267494345, '\U0009167aØ': -20643, '\U000a2c62': 4531811023802960318, '\U00093676Ö«\x1d¡\x85': -922}
dict({'\U0003817a': 18965, '': -77, '\x05\U0003a5d7': -16459}) = {'\U0003817a': 18965, '': -77, '\x05\U0003a5d7': -16459}
dict({'': -2515190650846871371, 'û-Æ(1î\x1a\x11®\U0008f30eÜ\U000aa817\U000fa00c~戯\x9e\x13': 21, 'g0¨x\rF': -113}) = {'': -2515190650846871371, 'û-Æ(1î\x1a\x11®\U0008f30eÜ\U000aa817\U000fa00c~戯\x9e\x13': 21, 'g0¨x\rF': -113}
dict({'': 21166, '\x9dÔt\U000fda81': -1852918762, '\tª\U00109142ð': 6878, 'ÿ6\x0c\xa0\U000ebc5f\x94\x7f': -21748, 'Ì\x8e': 9650, '\x02¦%\x99\U000e97ddR\U0007dc77Èü': -123, '\x16}': 1249, '\U000b480a': -24594, 'Ï': -20350, '\x16ã}Û\x8cf': -84, '\x95': -18322, '\x04': 20751, 'gÜ]\x8e«R¼Õû£\x1a®': 27656, '\x84': 88, 'Ol\U0005cc4b\U000e0a04Ä\x88\U000c30dc;$+à': -5507381615313056715, 'ÜÅ²': 3, '/A\x80Áp\x1f\x05': -5813, '±\n´`ßi\U000c002d\x04§J亘²\x9e@9i\x8crQ/¿\U000d1be8\U00085abcû«\n\U000f8270': 27061, '\U001082ddb\U0006f40c°z¸\x83\U000ba29d¹5\x84b\U00067e81*@b埙

ù\U0004306e\U000aaf53®½³ \x82\x1d': 58}) = {'': 21166, '\x9dÔt\U000fda81': -1852918762, '\tª\U00109142ð': 6878, 'ÿ6\x0c\xa0\U000ebc5f\x94\x7f': -21748, 'Ì\x8e': 9650, '\x02¦%\x99\U000e97ddR\U0007dc77Èü': -123, '\x16}': 1249, '\U000b480a': -24594, 'Ï': -20350, '\x16ã}Û\x8cf': -84, '\x95': -18322, '\x04': 20751, 'gÜ]\x8e«R¼Õû£\x1a®': 27656, '\x84': 88, 'Ol\U0005cc4b\U000e0a04Ä \x88\U000c30dc;$+à': -5507381615313056715, 'ÜÅ²': 3, '/A\x80Áp\x1f\x05': -5813, '±\n´`ßi\U000c002d\x04§J亘 ²\x9e@9i\x8crQ/¿\U000d1be8\U00085abcû«\n\U000f8270': 27061, '\U001082ddb\U0006f40c°z¸\x83\U000ba29d¹5\x84b\U00067e81*@b埙

ù\U0004306e\U000aaf53®½³\x82\x1d': 58}

dict({'µ]¬7Ù': 63, '%': 98, '9Ä\x8b¯*wZ\x0b': -4575}) = {'µ]¬7Ù': 63, '%': 98, '9Ä\x8b¯*wZ\x0b': -4575}

dict({'': -28991, '\U00105c70+î<JÌk)\x8f¯': -29075, 'Ø¥Á': 3422, '\U000e7ab8çZ>': -25693}) = {'': -28991, '\U00105c70+î<JÌk)\x8f¯': -29075, 'Ø¥Á': 3422, '\U000e7ab8çZ>': -25693}

dict({'': -31876497807671427, '\U00070c00î<JÌk)\x8f¯': -29075, 'Ø¥Á': 3422, '\U000e7ab8çZ>': -25693}) = {'': -31876497807671427, '\U00070c00î<JÌk)\x8f¯': -29075, 'Ø¥Á': 3422, '\U000e7ab8çZ>': -25693}

dict({'ùoÃ\U000b18aav例w\x04î@Çy\x10z ØÅ\U00050039\U0001aa8c\U000a4df5\U000c5d15LW': 13034299823944865623951801837996533654 3, '\x96\U00067e4e@Dô': -111}) = {'ùoÃ\U000b18aav例w\x04î@Çy\x10z ØÅ\U00050039\U0001aa8c\U000a4df5\U000c5d15LW': 13034299823944865623951801837996533654 3, '\x96\U00067e4e@Dô': -111}

dict({'\U0004a5f5\x07F\x88¡;4\U000a109c´[': -31518, '\x18\x89\x99\U00091491': -1153, '\U0003cca5\U000b6b9e': 106341153216707060812598426034269279050,

'1\U000dfc3e¯\U000d5be4ô\x18ÊÌ?\x8c\U0009389f': -
1483731776084758539795148875175676976 30, '': 920427709525766480,
'\x15¿\x82ÏO·{': -8418}) = {'\U0004a5f5\x07F\x88¡;4\U000a109c´[': -31518,
'\x18\x89\x99\U00091491': -1153, '\U0003cca5\U000b6b9e':
1063411532167070608125984260342692790 50,
'1\U000dfc3e¯\U000d5be4ô\x18ÊÌ?\x8c\U0009389f': -
1483731776084758539795148875175676976 30, '': 920427709525766480,
'\x15¿\x82ÏO·{': -8418}
dict({'': 26494, '\x0b\x80\x148': -14033, '\U000b6a3d': -15074,
'\U000cfce2\U000f3ab8\U0008496e>': 113, 'ô#\x97&\x8bÓÄ
t\x13=\U0010bca5\U000ceb48\n²□¹IQ @W\U00014335C\x12^ø17': 10493}) = {'':
26494, '\x0b\x80\x148': -14033, '\U000b6a3d':
-15074, '\U000cfce2\U000f3ab8\U0008496e>': 113, 'ô#\x97&\x8bÓÄ
t\x13=\U0010bca5\U000ceb48\n²□¹IQ @W\U00014335C\x12^ø17
': 10493}
dict({'\U000ef373': 20222, '椅\x1a\x94': 31206, '\U000418bb\U00086273*\x977':
14392, 'hE': -119, 'qb¶)\x85º': 7975, '\x94\U000d382d\x9f³\x9f\U000a8df6': 1, '':
5041, '¢岛/àI\U000c1ac5\\': 1137618700632669785, 'Zý¹Â\U00065fa4LÍ': -28552,
'\U000bed90\U000160e4': -125}) = {'\U000ef373': 20222, '椅\x1a\x94': 31206,
'\U000418bb\U00086273*\x977': 14392, 'hE': -119, 'qb¶)\x85º': 7975,
'\x94\U000d382d\x9f³\x9f\U000a8df6': 1, '': 5041, '¢岛/àI\U000c1ac5\\':
1137618700632669785, 'Zý¹Â\U00065fa4LÍ': -28552, '\U000bed90\U000160e4': -
125}
dict({'': -10656}) = {'': -10656}
dict({'': -13799}) = {'': -13799}
dict({'1\x03\U0009ae6d\ue18bþ¼5\U000eae7f': -7669850803142303404, 'ÒX\x90':

-26302, 'Í疚\x19ÿB3¤t\U0005e16c\x17\U0006c56eT4': 121, '\U000e28d0': 9993, '\U00070672': 8140, '': 171965912, 'ô\x12¥Ü\x1d\U00081146úöÄ': -742861556, 'ªSLi\U000b0d43ù': 24025, '¥': -115, '\U000dd5e3': -21029}) = {'1\x03\U0009ae6d\ue18bþ¼5\U000eae7f': -7669850803142303404, 'ÒX\x90': -26302, 'Í疚\x19ÿB3¤t\U0005e16c\x17\U0006c56eT4': 121, '\U000e28d0': 9993, '\U00070672': 8140, '': 171965912, 'ô\x12¥Ü\x1d\U00081146úöÄ': -742861556, 'ªSLi\U000b0d43ù': 24025, '¥': -115, '\U000dd5e3': -21029}

dict({'\x85z\U0003a0fdÓ\U0008a04bÄ6\U000c85d3Ýø': 29822, '': -7714, '\n\x08A¦\x1e': -4402860114928938884}) = {'\x85z\U0003a0fdÓ\U0008a04bÄ6\U000c85d3Ýø': 29822, '': -7714, '\n\x08A¦\x1e': -4402860114928938884}

dict({'\U00098218¾·': -7}) = {'\U00098218¾·': -7}

dict({'95¾·': -7}) = {'95¾·': -7}

dict({'95¾¾': -7}) = {'95¾¾': -7}

dict({'95¾¾': 0}) = {'95¾¾': 0}

dict({'95¾¾2': 72}) = {'95¾¾2': 72}

dict({'95¾¾2¾': 18432, '\U000e5ca3\x104B\n': 14, '\x8a»': -1836, '`': 8905}) = {'95¾¾2¾': 18432, '\U000e5ca3\x104B\n': 14, '\x8a»': -1836, '`': 8905}

dict({'95¾¾2¾\x10\x90': -768, '□14B\n': 14, '\x8a»': -1836, '`': 8905}) = {'95¾¾2¾\x10\x90': -768, '□14B\n': 14, '\x8a»': -1836, '`': 8905}

dict({'`': -15110, '\U00042a12': 26105, '\U000eb046»\x83b©îaó/mÙV3\U000528ce': -1715, '¡': -57, '': 19442, '  .\x01\U000a795b'}í": 350353607, '\x97ú½jnH\U000a4decD\U000849fc\U000929a0c\x06\U000f026e\xa0ôO': -1754174494110077180}) = {'`': -15110, '\U00042a12': 26105, '\U000eb046»\x83b©îaó/mÙV3\U000528ce': -1715, '¡': -57, '': 19442, ' .\x01\U000a795b'}í": 350353607, '\x97ú½jnH\U000a4decD\U000849fc\U000929a0c\x06\U000f026e\xa0ôO': -

1754174494110077180}