

Bashshell Basic Commands

gedit example.txt create a new
& empty file

1.1 Bashshell Directory Manipulation

pwd	Name of the current directory
cd dir	change the directory to c d
cd ~	Change to home directory
cd ..	Change to parent directory
cd .	Change to current directory
mkdir dir	Creates a new directory
rmdir dir	removes a directory
rm -rf dir	Removes non-empty directory dir and all of its subdirectories and files
mv dir1 dir2	Rename dir1 to dir2

1.2 Bashshell File Manipulation

1.3 Bash Shell Searching

find . -name "hello.txt" -type
file

find . -name "hello.*" -type
file

find . -name "hello" -type d

find . -name "*.c"

find . -name "*.ch"

grep '^apple' example

grep 'apple\$' example

grep '^apple\$' example

grep 'p?' example

grep 'p+' example

grep 'p*' example

grep 'p.' example

'[12345]' or '[1-5]'

(ab){3}

1.4 Useful Bash Shell Commands (cont)

Search by file name for a file	sort -n file	numerical sort
Search any file	sort -n -r file	-r for reverse sort
type with the name "hello"	sort -k3 -n filename	sort by field number
Search for a directory called "hello"	sort -t, -k3 -n filename	sort by field number but the delimiter is comma
find .c files	cut -d, -f 1,3 example1	return first and third column where delimiter is a space
find .c and .h files	diff file	remove adjacent duplicate lines
match beginning of a line	diff fileA fileB	how to transform fileA to fileB
match the end of a line		add the line1 of fileB after line0 of fileA
exact contents of a line		change line2,3 of fileA to line3 of fileB
match zero or one occurrence		display incorrect words in file
match one or more occurrences		change super user mode
match zero or more occurrences		install program
match a single character "p"		
match any character enclosed by []		
3 occurrences of "ab"		

1.5 Standard I/O and pipe

gedit hello.cpp &	Starts the gedit program to edit file hello.cpp	(ab){1,3}	1 to 3 occurrences of "ab"
g++ hello.cpp -o hello.o	compile hello.cpp to create executable a.o	(ab){3,}	3 or more occurrences of "ab"
./a.o	invoke a.o	grep -c " UNIX X" bar.txt	no. of occurrence for "UNIX"
cp file1 file2	Copy file1 into file2	sed s/UNIX /Unix/g bar.txt	replace all "UNIX" to "Unix"
cp -r dir1 dir2	Copy dir1 (including its subdirectories) to dir2	Note that (ab) can also be other valid patterns	
mv file dir	moves file to dir	1.4 Useful Bash Shell Commands	
rm file	remove file	wc -c file	return no. of bytes
rm dir	remove all files and directories in dir	wc -l file	returns no. of lines
touch file	Create an empty file called file	wc -w file	returns no. of words
cat file	Display the content of file	sort file	sort file in alphabetical order and output the sorted texts
		ls -l grep "Jan 25"	

File Security and Permission

ls -lt print in long form with security level indication

chmod [who] [operator] [permissions] filename

[who] u = user; g = group; o = other; a = all

[operator] + = add; - = remove; = set

[permission] r = read; w = write; x = execute



By Jack84
cheatography.com/jack84/

Not published yet.
 Last updated 17th December, 2018.
 Page 1 of 11.

Sponsored by [CrosswordCheats.com](https://crosswordcheats.com)
 Learn to solve cryptic crosswords!
<http://crosswordcheats.com>

C user input

```
int a; float b;
scanf( " %d% f", &a, &b);
printf ("%g ", a*b);
```

C++ const keyword

```
BigInteger add(const BigInteger
& a, const BigInteger & b)
{
    // Cannot modify any of the
    parameters
}

string BigInteger::setNumber(
string number) const {
    // setNumber is a read-only
    function.
    sign = "-"; // error because
    sign is a member variable of
    BigInteger
}
```

C++ Constructor

```
class Point{
private:
    int x, y;
public:
    Point() { // default
        a = 10; b=20
    }
    Point(int x1, int y1){
        // parameterised
        x = x1; y = y1;
    }
}

int main(){
    Point p; // default
    Point q(10, 20); // parameterised
}
```

Constructor is a member function that shares the same name as the class

C++ friend

C++ friend (cont)

```
> // add declaration of friend function
AFTER the declaration of the big integer
class
// --- in .cpp file --- //
BigInteger add(BigInteger a, BigInteger {
    // implement the function
}
```

2.1 Shell Script basics

```
echo -n " hello world" print without
" \n
```

```
read name read user
input and
store it in a
var called "-
name"
```

```
a=apple no quote
a='apple pie' single quote
(strings)
```

```
a="$ a\$ " can handle
special
characters
```

```
$ variable
substitution
```

```
\ escape
special
characters
```

```
`cat hello.txt` enclose bash
commands
```

```
a=`wc -l file | cut -d\" \" -f1`
"
```

```
echo " there are $a lines in file
"
```

2.2.1 Shell Script - Using Strings

```
${#a} length of string
${a:pos:len} substring (assume index 0)
${a:from/to} change part of string
a="Apple pie"; from="pie"; to="juice";
```

2.2.4 Shell Scripting - Variable as numbers

```
let " a=$ a+1 " increment a by 1
```

2.3 Shell Scripting - Control Flow

2.3 Shell Scripting - Control Flow (cont)

```
> echo "Do you want to remove all .cpp files
(Y/N)"
read ans
if [ "$ans" == "Y" ]
then
    rm -rf *.cpp
    echo "All .cpp files are removed"
fi
# --- for-in loop --- #
#!/bin/bash
list="1 2 3 4 5"
for i in $list
do
    echo "This is iteration $i"
done
# --- for-loop with a range --- #
for ((i=0; i<=100; i=i+3))
do
    echo $i
done
```

2.4 Shell Scripting - Useful techniques

```
$# number of arguments input by
user
$0; $1; $2 1st argument, 2nd argument,
cp file123 fileabc 1> /dev/null 2>
/dev/null is system dust bin
&1 is the standard output
echo " $0: error:Copy failed " >&
&2 is error output
```

Shell Script Conditions

String comparison

```
[ " $s " ] iff length of s is
non-zero
[ " s1 " == " $s2 " ]
[ " $s1 " != " $s2 " ]
[ " $s1 " /> " $s2 s1 sorted after
" ] s2
[ " $s1 " /< " $s2 " ]
```

File Checking

```
[ -e $file ] iff exists
[ -f $file ] iff is a file
[ -d file ] iff is a directory
```

```
// --- In .h file ---
class BigInteger{
public:
    void setNumber( string );
    string getNumber();
private:
    char sign;
    int length;
    int value[ 100];
    friend BigInteger add(BigInteger a, BigInteger b);
};
BigInteger add(BigInteger a,
BigInteger b);
```

```
# --- if-else statements --- #
if [ condition ]
then
    echo " Action 1"
elif [ condition2 ]
then
    echo " Action 2"
else
    echo " Action neither"
fi

#example
#!/bin /bash
```



By Jack84

cheatography.com/jack84/

Not published yet.

Last updated 17th December, 2018.

Page 2 of 11.

Sponsored by CrosswordCheats.com

Learn to solve cryptic crosswords!

<http://crosswordcheats.com>

Shell Script Conditions (cont)

Number Comparison

[\$a -eq \$b]	iff a = b
[\$a -ne \$b]	iff a!=b
[\$a -lt \$b]	iff a<b
[\$a -le \$b]	iff a<=b
[\$a -gt \$b]	iff a>b
[\$a -ge \$b]	iff a>=b

Iterate words in a file

```
list=`cat wordlist.txt`
for line in $list
do
    echo " $li ne"
done
```

C++ Misc

```
void func(int array[]); // array as
parameter
```

C++ Dynamic Array

```
int * a = NULL; int n;
cin >> n; a = new int[n];
...
delete[] a; // free memory
```

Pointers C++

int *baz	define a pointer
&foo	address of foo
*baz	value pointed by baz
void PBV(int *p)	parameter is a pointer
void PBR(int *&p)	parameter is address of pointer

modifying the parameter modifies the original variable

C++ Linked List and Various functions

C++ Linked List and Various functions (cont)

```
> while (current!=NULL){
    cout << "Key:" << current->key <<
",value:" << current->value << endl;
    current = current->next;
}
}
bool isSorted(Node *head){
    Node *current = head;
    Node *previous = NULL;
    while (current!=NULL){
        if (previous !=NULL){
            if (previous->key > current->key)
                return false;
        }
        previous = current;
        current = current->next;
    }
    return true;
}
void insertInOrder(Node *&head, int k){
    Node *newNode = new Node;
    newNode->key = k;
    if (head == NULL)
    {
        newNode->next = NULL;
        head = newNode;
    }
    else
    {
        Node *current= head;
        Node *previous = NULL;
        while(current!=NULL)
        {
            if (current->key > k)
                break;
            previous = current;
            current=current->next;
        }
        newNode->next = current;
        if (previous!=NULL)
            previous->next = newNode;
        else
            head = newNode;
    }
}
```

C++ Linked List and Various functions (cont)

```
> }
}
```

4. Separate Compilation and Makefile

```
census.o: census.cpp
BigInteger.h Country.h
g++ -c census.cpp
BigInt ege r.o :Bi gIn teger.h
BigInt ege r.cpp
g++ -c BigInt ege r.cpp
Countr y.o :Bi gIn teger.h
Country.h Countr y.cpp
g++ -c Countr y.cpp
census :ce nsus.o BigInt eger.o
Country.o
g++ census.o BigInt eger.o
Country.o -o census
```

C++ Traverse Linked List using for-loop

```
void TraverseList (Node *head) {
    for (Node *n = head; n ->
next != NULL; n = n -> next)
    { //do something to n }
}
```

C++ Operator overloading

```
// Using friend functions
BigInteger operat or+ (const
BigInteger &a, const BigInteger
&b);
istream &o perator >> (istream
&cin, BigInteger &b);
ostream &o perator << ostream
&cout, BigInteger &b);
```

C AVL Tree - Node and maximum

```
struct treeNode {
    int key; struct treeNode *
left;
    struct treeNode * right;
}
typedef struct treeNode
treeNode;
int maximu m(int a, int b){
    return (a > b ? a : b);
}
```

C AVL - Rotation Functions

```

int main(){ Node *head = NULL;
... }
void headInsert(Node *& head,
int k, int v){
    Node *newNode = new Node;
    newNode -> key = k;
    newNode -> value = v;
    newNode -> next = head;
    head = newNode;
}
void printList (Node *head){
    Node *current = head;

```

```

treeNode* R_rotation(treeNode
*parent){
    treeNode *child =
parent -> left;
    parent -> left = child -
> right;
    child -> right = parent;
return child;}
treeNode* L_rotation(treeNode
*parent){
    treeNode *child =
parent -> right;
    parent -> right = child
-> left;

```



By **Jack84**

cheatography.com/jack84/

Not published yet.

Last updated 17th December, 2018.

Page 3 of 11.

Sponsored by **CrosswordCheats.com**

Learn to solve cryptic crosswords!

<http://crosswordcheats.com>

C AVL - Rotation Functions (cont)

```
> child -> left = parent; return child;}
treeNode* LR_rotation(treeNode *parent){
    treeNode *child = parent -> left;
    parent -> left = L_rotation(child);
    return R_rotation(parent);}
treeNode* RL_rotation(treeNode *parent){
    treeNode *child = parent -> right;
    parent -> right = R_rotation(child);
    return L_rotation(parent);}
```

C AVL - Insert()

```
treeNode* Insert(treeNode
*currentNode, int key){
    if( cur ren tNode == NULL){
        cur ren tNode = (treeN -
ode *)m all oc( siz eof (tr eeN -
ode));
        cur ren tNode -> key =
key;
        cur ren tNode -> left =
curren tNode -> right = NULL;
    }
    else if(key > curren tNode ->
key){
        cur ren tNode -> right =
Insert (cu rre ntNode -> right,
key);
        cur ren tNode = balanc -
e_t ree (cu rre ntN ode);
    }
    else if(key < curren tNode ->
key){
        cur ren tNode -> left =
Insert (cu rre ntNode -> left,
key);
        cur ren tNode = balanc -
e_t ree (cu rre ntN ode);
    }
    else {
        pri ntf ("fail! -
duplicated key \n");
        exi t(-1);
    }
    return curren tNode;
}
```

C AVL - get_height

C AVL - get_height (cont)

```
>}
```

C AVL - getBalance()

```
int get_balance(treeNode *
currentNode){
    if( cur ren tNode == NULL)
return 0;
    else return get_he igh t(c -
urr ent Nod e-> left) - get_he -
igh t(c urr ent Nod e-> hei -
ght);
}
```

C AVL - balance_tree()

```
treeNode* balance_tree(treeNode
* currentNode){
    int height_diff = get_ba -
lan ce( cur ren tNode);
    if( hei ght _diff > 1)
    {
        if( get _ba lan ce( -
cur ren tNode -> left) > 0){
            cur ren tNode =
R_rota tio n(c urr ent Node);
        } else {
            cur ren tNode =
LR_rot ati on( cur ren tNode);
        }
    } else if(hei ght _diff < -1)
    {
        if( get _ba lan ce( -
cur ren tNo de->right) < 0){
            cur ren tnode = L_rota -
tio n(c urr ent Node);
        } else {
            cur ren tNode =
RL_rot ati on( cur ren tNode);
        }
    }
    return curren tNode;
}
```

C AVL - main()

```
int main(){
    tre eNode *root = NULL; root
= Insert (root, 5);}
```

5.1 Containers

5.1 Containers (cont)

l.pop_ front(	remove the first item
)	
l.pop_ back()	remove last item
l.front()	access the first item
l.back()	access the last item
l.size()	return num of items
map<K, V> m;	map definition
m[i]	i-th item in the list
m.count(k)	return no. of pairs in the map with key = k
m.size()	no. of items

Using map with user define objects

```
bool operator<(const Record& a,
const Record& b){
    return a.name < b.name;
}
```

Must overload "<" operator

Directives for STL

```
#include<vector>;
#include<list>; #include<map>;
#include<algorithm>;
```

Algorithms

```
int get_height(treeNode
*currentNode)
{
    if( currentNode == NULL)
        return 0;
    else{
        int height = 1 + maximum -
m(get_height (currentN -
ode->left), get_height(c -
urrentNode->right));
        return height;
    }
}
```

<code>vector <int></code>	vector definition
<code>v;</code>	
<code>v[i]</code>	i-th item in the vector
<code>v.pop_back()</code>	remove last item
<code>v.size()</code>	size of vector
<code>list<int> l;</code>	list definition
<code>l.push_front()</code>	insert item at front
<code>l.push_back()</code>	insert item at back

Sorting

`sort(v.begin(), v.end()) // vector`

`sort(a, a+10); // array`

`c.sort() // list and maps`

`sort(v.begin(), v.end(), compare`

`bool compare(int a, int b){ return`

`overload operator<() for special tricks`

Binary Search

`binary_search(v.begin(), v.end()`
`bool`

"target" is what you are looking for in v

Upper & lower bound

`lower_bound(v.begin(), v.end(), target); // return`

`upper_bound(v.begin(), v.end()`

`lower_bound()` returns the earliest position

`upper_bound()` returns the latest position

`binary_search(), upper_bound()` are
used with vectors, lists, and maps

Random Shuffle (see appendix)

`need <cstdlib>, <ctime>, srand( time`



By Jack84

cheatography.com/jack84/

Not published yet.

Last updated 17th December, 2018.

Page 4 of 11.

Sponsored by **CrosswordCheats.com**

Learn to solve cryptic crosswords!

<http://crosswordcheats.com>

C++ STL Template Class

```
template <class T>
class MyCollection{
    vector <T> data;
public:
    void Add(const T &);
};

template <class T>
void MyCollection <T> ::
Add(const T & d){
    data.push_back(d);
}
```

C++ Template Operator Overloading (One-to-one)

```
template <class T>
class MyCollection{
    vector <T> data;
public:
    void Add(T const &);
    T & Draw();
    friend ostream & operator -
or< <(ostream & cout, const
MyCollection <T> &q){
    cout << " Collection"
<< endl;
    typename vector -
<T> ::const_iterator itr;
    for(itr=q.data -
begin(); itr != q.data.end();
itr++)
        cout << " " <<
*itr << endl;
    return cout;
}
```

C++ Template Overloading (many-to-many)

C++ Template Overloading (many-to-many) (cont)

```
> typename vector<U>::const_iterator itr;
... (same)
}
```

C Conversion Specifier

int	%d
float	%f
double	%lf
char	%c
string	%s

C string

```
char name[] = "Alan";
char name[100]; scanf( "%s", name);
#include<string.h> more functions
```

```
strcpy(char s1[], char s2[])
copy s2 to s1
```

```
strcat(char s1[], char s2[])
append s2 to end of s1
```

```
strcmp(char s1[], char s2[])
return -1 if s1<s2, 0 if s1==s2, 1 if s1>s2
```

```
strlen(char s1[])
return length of string
```

C Functions

Pass by reference

```
void swap(double *a, double *b){...}
}
```

Using this function: pass the address

```
swap(&a, &b);
```

C Arrays

```
int array[] = {1, 2, 3};
a[i] is the same as *(a+i)
```

C Memory Allocation

Python

```
int(a); float(a); str(a)
type casting
```

```
a+b
concatenation
```

```
s[i]
access i-th
```

```
s[1:5]
substring s[1] to s[4]
```

```
s[1:]
s[1] to end
```

```
s[:4]
start to s[3]
```

```
len(s)
length of the string
```

```
print "howdy!",
print without newline
```

```
s = input("prompt")
take user input as strings
```

Python File Input

```
with open("filename", "mode")
as f:
```

```
f.close()
```

```
s = f.read();
```

```
return 0
if s1==s2: write(s tr(a));
```

file modes

```
r
```

```
r+
```

```
w+
```

```
a
```

```
a+
```

```
template <class T>
class MyCollection{
    vector<T> data;
public:
    void Add(T const &);
    T & Draw();
    template <class U>
    friend ostream & operator<
or< <(ostream & cout, const
MyCollection <U> & q);
};
template <class U>
ostream & operator< <(ostream
&cout, const MyCollection <U>
& q)
{
```

```
int size; int *a; scanf( " %d", &size);
e);
a = malloc (size* sizeof (int));
free(a); // a is pointer
```

C Structure and typedef

```
struct student { char name[20];
int uid; }
typedef struct student student;
int main(){ student a; ... }
```

Python Flow of Control

#if-else statement

```
if condition:
    statement
elif condition:
    statement
else:
    statement
```

Conditions are not enclosed by brackets

C

By Jack84

cheatography.com/jack84/

Not published yet.

Last updated 17th December, 2018.

Page 5 of 11.

Sponsored by CrosswordCheats.com

Learn to solve cryptic crosswords!

<http://crosswordcheats.com>

Python Logical operators

&&	and
	or
!	not

Python For-loops

```
for i in list:
    sta tement
    sta tement
```

#Example 1

```
i = 1
for dir in [ " n", " e", " w",
" s" ]:
    print " the " + str(i) + "-
th direction is" + dir
    i+=1
```

#Example2

```
for i in range(0, 71):
    if i % 7 == 0:
        print i,
```

Python Array

```
arr = [0] * i    empty array with size i
arr[i]           i-th item in array
```



By Jack84

cheatography.com/jack84/

Not published yet.

Last updated 17th December, 2018.

Page 6 of 11.

Sponsored by CrosswordCheats.com

Learn to solve cryptic crosswords!

<http://crosswordcheats.com>