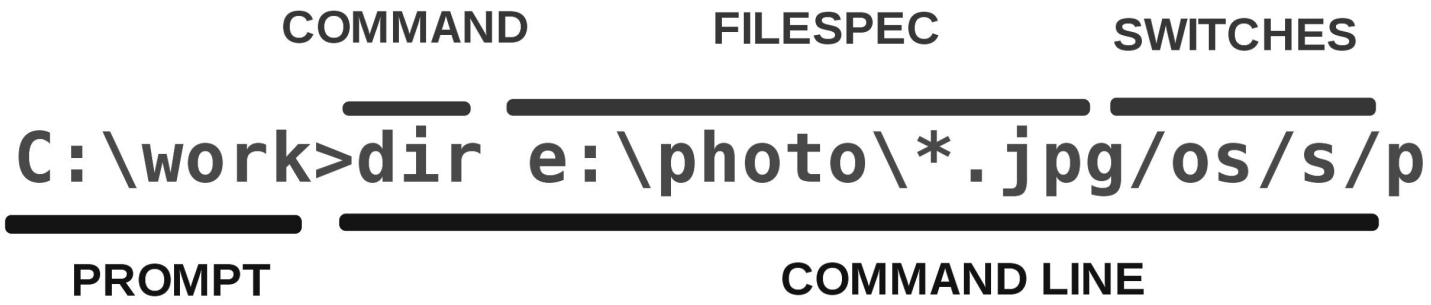
CS481/CS583: Bioinformatics Algorithms

Can Alkan

http://www.cs.bilkent.edu.tr/~calkan/teaching/cs481/



Short note on homework implementations

IMPLEMENTING ARGUMENT PARSER

C: getopt.h

- When you need something like:
  - `myprog –input input.fasta`
  - `myprog –search input.fasta –q query.fasta`
  - `myprog –search input.fasta –q query.fasta –max-hits 10`
- Use getopt library
- Enumerators are useful

C: getopt.h

```
#include <getopt.h>

int getopt_long(int argc,
                char * const argv[],
                const char *optstring,
                const struct option *longopts,
                int *longindex);
```

struct option

- `const char *name`: name of the option
- `int has_arg`: one of required_argument, no_argument, optional_argument
- `int *flag`
- `int val`

If `flag` is a *null pointer*, then the `val` is a value which identifies this option.

http://man7.org/linux/man-pages/man3/getopt.3.html

Example: set up arguments

```
enum modes {NONE, INDEX, SEARCH};

int main(int argc, char **argv){

     enum modes mode;
     int index;
     int o;
     mode = NONE;
     static struct option long_options[] =
     {
               {"input", required_argument, 0, 'i'},
               {"search", required_argument, 0, 's'},
               {"query", required_argument, 0, 'q'},
               {0, 0, 0, 0}
     };
```

Example: parse

```
while(
     (o = getopt_long( argc, argv, "i:s:q", long_options, &index)) != -1
     )
{
     switch (o) {
          case 'i':
               strcpy(filename, optarg);
               break;
          case 's':
               mode = SEARCH;
               strcpy(filename, optarg);
               break;
          case 'q':
               strcpy(queryfile, optarg);
               break;
     }
}
```

Example: validity check

```
if (mode == NONE){
     fprintf(stderr, “Use either –index or –search\n”);
     return 1;
}

if (mode == SEARCH && queryfile[0] == 0) {
     fprintf(stderr, “Cannot search without a query\n”);
     return 1;
}
```

C++: argh.h

```cpp
#include <iostream>
#include "argh.h"

int main(int argc, char *argv[])
{
    argh::parser cmdl;
    cmdl.parse(argc, argv, argh::parser::PREFER_PARAM_FOR_UNREG_OPTION);

    if (cmdl["-v"]) // check for flags with []
        std::cout << "verbose enabled." << std::endl;
    std::string i, f;
    cmdl("-i") >> i; // get values of args with ()
    cmdl("-f") >> f;
    std::cout << "-i:" << i << std::endl std::cout << "-f:" << f << std::endl;
    return 0;
}
```

https://github.com/rica01/cli-args

C++: argh.h

What happens if a flag is not present?

No problem.

```
std::string na; // not in list!
cmdl("-na") >> na; // empty string
std::cout << "-na:" << na << std::endl;
```

1.Compiling

```
g++ main.cpp
```

2.Running

```
./a.out -i III -f FFF -q ???
```

3.Output

```
-i:III
-f:FFF
```

https://github.com/rica01/cli-args

Python3: argparse

```
import argparse

parser = argparse.ArgumentParser(description='Basic calculator')
parser.add_argument('--i', type=str, help='input file')
parser.add_argument('--f', type=str,  help=config file')

args = parser.parse_args()

print(args.i)
print(args.f)
```

Python3: `argparse`

1.Compiling

```
no need! Python!
```

2.Running

```
python3 a.py -i III -f FFF
```

3.Output

```
-i:III
-f:FFF
```

Java: cli-args

```
public class Demo {

    public static void main(String[] argv) {

        CliArgs cliArgs = new CliArgs(argv);

        double n = cliArgs.switchDoubleValue("-n");
        double f = cliArgs.switchValue("-f");
        String i= cliArgs.switchValue("-i");

        System.out.println("n:" + n.toString());
        System.out.println("f:" + f.toString());
        System.out.println("i:" + i);
        return;


    }
}
```

https://github.com/rica01/cli-args

Java: cli-args

1. Compiling

```
javac Demo.java
```

2. Running

```
java Demo -n 3.2 -i III -f FFF
```

3. Output

```
-n 3.2
-f:FFF
-i:III
```

https://github.com/rica01/cli-args

Tutorial

https://www.youtube.com/watch?v=_-ydayA-S9M

https://tinyurl.com/cs481argparse