Benjamin Kaplan
Problem set 4
==========

```
binyamin@BenjaminButtox:~/Documents$ ./wordgen 1000 | ./wordsearch words.txt | ./pager
ZEE
PYX
KHU
JIS
...
TIT
ATTL
SYU
SAR
YEH
DOV
VOW
GUY
--Pess RETURN for more--

YUN
CAE
RLE
35 words matched
binyamin@BenjaminButtox:~/Documents$ ./wordgen 10000 | ./wordsearch words.txt | ./pager
ZEE
PYX
KHU
...
TIT
ATTL
SYU
SMR
YEH
DOV
GUY
--Pess RETURN for more--

YUN
CAE
...
RLE
NIY
UNU
SME
ATU
WYC
HOG
RAN
--Pess RETURN for more--
Q
***Pager terminated by Q***
156 words matched

binyamin@BenjaminButtox:~/Documents$ gcc testLauncher.c
binyamin@BenjaminButtox:~/Documents$ ./a.out 1000
POE
RAN
PHARE
CHY
SON
FEE
...
FID
YVO
KYT
PUB
```

```
PST
--Pess RETURN for more--

KOGA
CGS
HEC
...
BEMA
VOB
HIAN
UGHT
IBY
VOI
44 words matched



=========================================================
Source Code:
=========================================================
wordgen:

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <errno.h>
#include <fcntl.h>
#include <sys/stat.h>
#include <sys/types.h>
#define MAXCHARS 7
int main(int argc, char **argv){
  int numWords = 0;
  if(argc>1)
    numWords = atoi(argv[1]);
  char *letters[26] = {"A","B","C","D","E", "F", "G", "H", "I", "J", "K", "L",
                      "M", "N", "O", "P", "Q", "R", "S", "T", "U", "V", "W",
                      "X","Y", "Z"};
  int fd;
  int wordsMade = 0;
  srand(time(0));
  do{
    if(numWords != 0)
      wordsMade++;
    else
      wordsMade--;
    double random = (MAXCHARS+1)*(double)rand()/(double)RAND_MAX;
    int length = (int)random+3;
    int i = 0;
    char *word = malloc(64);
    int randomInt;
    char* letter;
    for(i = 0; i<length; i++){
      randomInt =  (26)*(double)rand()/(double)RAND_MAX;
      letter = letters[randomInt];
      strcat(word, letter);
    }
    printf("%s\n", word);

  }while(wordsMade < numWords);

return 0;

}

=========================================================
wordsearch:
```

```
#include <string.h>
#include <stdio.h>
#include <errno.h>
#include <stdlib.h>
#include <fcntl.h>
#include <setjmp.h>
#include <sys/signal.h>
int numMatched = 0;

void pipe_handler(int sn){
  printf("%d words matched\n", numMatched);
}


int main(int argc, char **argv){
  signal(SIGPIPE, &pipe_handler);

  if(argc !=2){
    fprintf(stderr, "Error: Wrong number of arguments\n");
    return -1;
  }


  char *word = NULL;
  FILE *fp;
  if( !(fp = fopen(argv[1], "r")))
    fprintf(stderr, "Error: %s, Errno: %d\n", strerror(errno), errno);
  size_t size = 0;
  char *input = NULL;

  int num = 370098 * sizeof(char*);
  int matched = 0;

  int z = 0;

  int a = 0;
  while(0<(a = getline(&input, &size, stdin))){

    for(z = 0; z<370098; z++){
      getline(&word, &size, fp);

      if ( !strncmp(input, word, strlen(input))){
        matched =1;
      }
    }
    rewind(fp);
    if( matched == 1){
      matched =0;
      printf("%s", input);
      numMatched++;

    }
    //getline(&input, &size, stdin);

  }
  printf("%d words matched\n", numMatched);

  return 0;
}

==================================================
pager:

#include <string.h>
#include <stdio.h>
#include <errno.h>
#include <stdlib.h>
```

```
#include <fcntl.h>

int main(int argc, char **argv){
  int lineNum = 0;
  char *line;
  size_t size = 0;
  char c = 0;
  FILE *fp;
  if(!(fp = fopen("/dev/tty", "r"))){
    fprintf(stderr, "Error: %s, Errno: %d\n", strerror(errno), errno);
    return -1;
  }

  int j = 0;

  while( (c!=81) && (c!= 113) && (c!= EOF)){
    for(j = 0; j < 23; j++){
      if(0 > getline(&line, &size, stdin))
        return 0;
      printf("%s", line);
    }
    printf("--Pess RETURN for more--\n");
    c = fgetc(fp);

  }
  if ((c ==81) ||(c==113))
    printf("***Pager terminated by Q***\n");
  return 0;
}
============================================================
Launcher:

//Benjamin Kaplan- PS 4
// launcher.c

#include <errno.h>
#include <string.h>
#include <stdlib.h>
#include <fcntl.h>
#include <sys/signal.h>
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>

int main(int argc, char **argv){
  char* secondParam = "0";
  if(argc==2)
    secondParam = argv[1];
  void errorA(){

      perror("Error: ");
      fprintf(stderr, " Errno: %d\n", errno);

      exit -1;

  }
  int pipefd1[2] = {10,11};
  int pipefd2[2] = {12,13};
  if(pipe(pipefd1)<0)
    errorA();


  pid_t PID1 = -1;
  pid_t PID2 = -1;
  pid_t PID3 = -1;
  if((PID1 = fork())<0)
    errorA();
```

```
if(PID1 == 0){
  printf("in child wordgen\n");
  if( dup2(pipefd1[1],1)< 0)
    errorA();

  if ((close(pipefd1[0]) || close(pipefd1[1])) <0){
    errorA();
  }
  printf("%s\n", argv[1]);
  if(execl("./wordgen", "./wordgen", secondParam, (char *) NULL)<0)
    errorA();
}
if(PID1 != 0){
  if (pipe(pipefd2)<0){
    errorA();
  }

  if( (PID2 = fork())<0){
    errorA();
  }
  if(PID2 == 0){

    if(dup2(pipefd1[0],0)<0)
      errorA();
    if(close(pipefd1[0])<0)
      errorA();
    if(close(pipefd1[1])<0)
      errorA();
    if (dup2(pipefd2[1],1)<0){
      errorA();
    }
    if(close(pipefd2[0])<0)
      errorA();
    if(close(pipefd2[1])<0)
      errorA();
    printf("in child wordsearch\n");
    if(execl("./wordsearch", "./wordsearch", "words.txt", (char *) NULL)<0)
      errorA();
  }
  if(PID2 != 0){

    if((PID3 = fork())<0){
      errorA();
    }
    if(PID3 == 0){
      if(dup2(pipefd2[0],0)<0)
        errorA();
      if(close(pipefd2[0])<0)
        errorA();
      if(close(pipefd2[1])<0)
        errorA();
      printf("in child pager\n");
      if(execl("./pager", "./pager", (char *) NULL )<0)
        errorA();
    }
    if(PID3 != 0){
      if(close(pipefd2[0])<0)
        errorA();
      if(close(pipefd2[1])<0)
        errorA();
      int status3;
      if(wait(&status3)<0)
        errorA();
      printf("Child pager %d exited with %d\n", PID3, status3);
    }
    int status2;
```

```
      if(wait(&status2)<0)
        errorA();
      printf("Child wordsearch %d exited with %d\n", PID2, status2);

    }

    int status1;
    if(wait(&status1)<0)
      errorA();
    printf("Child wordgen %d exited with %d\n", PID1, status1);
  }

  return 0;
}
```