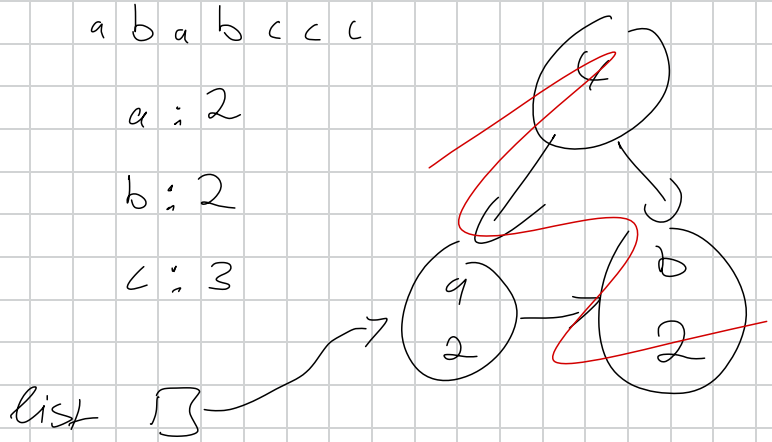
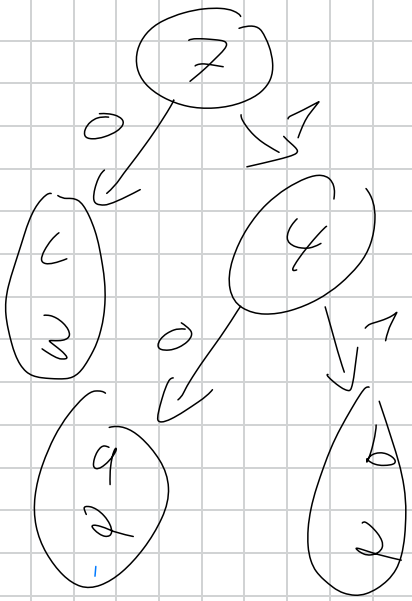
Input: a b a b c c c

a : 2

b : 2

c : 3



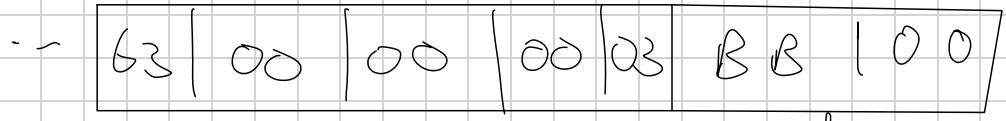


codes: a : 10

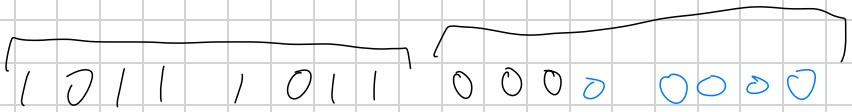
b : 11

c : 0

histogram



histogram Body



BB 00

a b a b c c c

```
#define HUNK 100

char *rll(FILE *whence){
  /* readline, returns line or NULL on EOF */
   * consumes new lines */
   char *s; int c;
   int size, len;

   size = HUNK;
   len = 0;
   if(!(s= malloc(size))){
     perror("malloc");
     exit(3);
   }
   while((c = getchar())!='\n' && c=EOF){

        if(len >= size-2){
          size += HUNK;

          s= realloc(s, size);

          if(!s){
             perror("realloc");
             exit(42);
          }
        }
        s[len++]=c;
   }
   s[len]='\0';
   if(c==EOF && len==0){
     free(s); -> s=NULL; -> }
   return s;
}

-> if(!(realloc(s,q,+1)))
   {perror("realloc was stupid");
    exit(EXIT_FAILURE);
```

```
int main(int argc, char *argv[]){
    char *old, *new;
    old = rll(stdin);

    while(old && (new=rll(stdin))){
        if(!strcmp(old,new)){ /*same*/
        }else{
            puts(old);
            free(old);
            old=new;
        }
    }
    if(old)
        puts(old);
    free(old);
    return 0;
}
```