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

void explode(const int code);

void readInput(char buffer[], const int bufferSize){
  int i;
  int c;
  for(i=0; i<=bufferSize; ++i){
    c = getchar();

    /* this checks that you are at the end of the line */
    /* Windows encodes an end−of−line as two characters: \r\n */
    /* Linux just uses \n */
    /* This will accept either version */
    if(c == ’\r’){
      c = getchar();
    }

    if(c == ’\n’){
      break;
    }
    else if(i<bufferSize){
      buffer[i] = (char)c;
    }
  }

}

void part1(){
  const int bufferSize = 12;
  char buffer[bufferSize];
  char target[] = "Careful";
  int i;
  const int n = strlen(target);

  readInput(buffer, bufferSize);

  for(i=0; i<n; i++){
    if(buffer[i] != target[i]){
      explode(1);
    }
  }
}

int part2(){
  int i,z;

  const int bufferSize = 12;
  char buffer[bufferSize];

  readInput(buffer, bufferSize);

  /* atoi takes a string representation of a number (e.g. "1234") */
  /* and converts it to an int representation of the number (e.g. 1234) */
  z = atoi(buffer);
  if(!z)
    explode(20);

  z += 17;

  if(z != 93)
    explode(21);

  return z / 2;
}

void part3(){
  int i,z;

  const int bufferSize = 12;
  char buffer[bufferSize];
```

```

  readInput(buffer, bufferSize);

  i=0;
  while(i < bufferSize && (buffer[i] == ’Z’ || buffer[i] == ’j’)){
    i++;
  }

  if(i != 3)
    explode(70);
}

void part4(const int x){
  int i,y,z;
  const int bufferSize = 12;
  char buffer[bufferSize];

  readInput(buffer, bufferSize);

  /* atoi takes a string representation of a number (e.g. "1234") */
  /* and converts it to an int representation of the number (e.g. 1234) */
  z = atoi(buffer);
  if(!z)
    explode(20);

  y = z;
  for(i=0; i<50; i++){
    y += z * i * y;
  }

  if(z + 9 != x){
    explode(21);
  }
}


void part5(char buffer[], const int bufferSize){
  const char allowed[] = "az3hoPeACce6";
  int i, j, x;

  for(i=0; i<bufferSize; ++i){
    char c;

    x = getchar();
    if(x == EOF)
      explode(10);
    else if(x == ’\n’)
      explode(11);

    c = (char)x;
    for(j=2; j<10; ++j){
      if(c == allowed[j]){
        buffer[i] = c + 1;
        break;
      }
    }
    if(j == 10){
      explode(12);
    }

  }

  j = 0;
  for(i=1; i<bufferSize; ++i){
    if(buffer[i] != buffer[i−1]){
      j = 1;
      break;
    }
  }
  if(j == 0)
    explode(13);

  /* this checks that you are at the end of the line */
  /* Windows encodes an end−of−line as two characters: \r\n */
  /* Linux just uses \n */
  /* This will accept either version */
  x = getchar();
  if(x == ’\r’)
    x = getchar();
  if(x != ’\n’)
    explode(14);
}


void part6(const char buf1[], const int bufSize){
  char buf2[bufSize];
  int i, c;

  readInput(buf2, bufSize);

  for(i=0; i<bufSize; ++i){
    if(buf1[i] != buf2[bufSize − 1 − i])
      explode(−44);
  }

}

void part7(int *x){
  int y = 29;
  int *z = &y;
  int k;

  const int bufSize = 12;
  char buffer[bufSize];

  *z −= 3;

  readInput(buffer, bufSize);

  /* atoi takes a string representation of a number (e.g. "1234") */
  /* and converts it to an int representation of the number (e.g. 1234) */
  k = atoi(buffer);

  if(k + *x != y)
    explode(70);

}

void theBomb(){
  const int bufSize = 10;
  char buffer[bufSize];
  int result;

  printf("Enter the stage 1 password: ");
  part1();
  printf("Stage 1 defused.\n");

  printf("Enter the stage 2 password: ");
  result = part2();
  printf("Stage 2 defused.\n");

  printf("Enter the stage 3 password: ");
  part3();
  printf("Stage 3 defused.\n");

  printf("Enter the stage 4 password: ");
  part4(result);
  printf("Stage 4 defused.\n");

  printf("Enter the stage 5 password: ");
  part5(buffer, bufSize);
  printf("Stage 5 defused.\n");

  printf("Enter the stage 6 password: ");
  part6(buffer, bufSize);
  printf("Stage 6 defused.\n");
```

```

  printf("Enter the stage 7 password: ");
  part7(&result);
  printf("Stage 7 defused.\n");

  printf("The entire bomb has been defused.\n");
}

```