**Module-2**

**Crossword Puzzle**

**#include <stdio.h>**

**#include <string.h>**

**#include <math.h>**

**#include <stdlib.h>**

**int fill\_puzzle(char \*puzzle, char words[10][10], int W, int \*lengths){**

**char start,oldentry[10],\*c;**

**int scpos = 0;**

**while(\*(puzzle+scpos)!='-'&& scpos<100) scpos++;**

**if(scpos==100)return 1;**

**int length = 0;**

**int across = 0;**

**int delta = 10;**

**if(\*(puzzle+scpos+1)=='-') {**

**across=1; //across clue**

**delta = 1;**

**}**

**while(\*(puzzle+scpos+delta\*length)!='+' && ((!across && (scpos+delta\*length<100)) || (across && (length<10\*((scpos/10)+1)-scpos)))) length++;**

**start = 0;**

**if((across && scpos%10>0) || (!across && scpos>10)){**

**if(\*(puzzle+scpos-across-(1-across)\*10)!='+') { // joined to another word**

**start = \*(puzzle+scpos-across-(1-across)\*10);**

**length++;**

**scpos -= across+(1-across)\*10;**

**}**

**}**

**for(int iw=0;iw<W;iw++){**

**int startok = 1;**

**if(start && start!=words[iw][0]) startok = 0;**

**if(lengths[iw]==length && startok) {**

**int temp = lengths[iw];**

**lengths[iw] = 0;**

**int fitok = 1;**

**for(int wc=0;wc<length;wc++) {**

**c = puzzle+scpos+10\*wc;**

**if(across) c = puzzle+scpos+wc;**

**oldentry[wc] = \*c;**

**if(\*c=='-'){**

**\*c = words[iw][wc];**

**} else {**

**if(\*c!=words[iw][wc])fitok=0;**

**}**

**}**

**if(fitok) fitok = fill\_puzzle(puzzle,words,W,lengths);**

**if(fitok) return 1;**

**lengths[iw] = temp;**

**for(int wc=0;wc<length;wc++) {**

**c = puzzle+scpos+across\*wc+(1-across)\*10\*wc;**

**\*c = oldentry[wc];**

**}**

**}**

**}**

**return 0;**

**}**

**int main() {**

**char puzzle[100], words[10][10], line[101], eol;**

**int \*lengths = calloc(10,sizeof(int));**

**for(int i=0;i<10;i++) {**

**scanf("%s",line);**

**for(int j=0;j<10;j++) \*(puzzle+10\*i+j) = line[j];**

**}**

**char c = 0;**

**int wpos = 0;**

**int nwords = 0;**

**scanf("%s",line);**

**for(int i=0;i<strlen(line);i++) {**

**if(line[i]==';') {**

**nwords++;**

**wpos = 0;**

**} else {**

**words[nwords][wpos] = line[i];**

**wpos++;**

**lengths[nwords]++;**

**}**

**}**

**nwords++;**

**int fitok = fill\_puzzle(puzzle,words,nwords,lengths);**

**for(int i=0;i<10;i++) {**

**for(int j=0;j<10;j++)printf("%c",\*(puzzle+10\*i+j));**

**printf("\n");**

**}**

**return 0;**

**}**

[**The Power Sum**](https://www.hackerrank.com/contests/module-2-1715051718/challenges/the-power-sum)

**#include <stdio.h>**

**#include <string.h>**

**#include <math.h>**

**#include <stdlib.h>**

**void recursive(int val, int p, int i, int max, int total, int \*count)**

**{**

**int new\_total = total + pow(i,p);**

**if(new\_total > val)**

**return;**

**if(new\_total == val)**

**{**

**\*count += 1;**

**return;**

**}**

**if(i > max)**

**return;**

**int j;**

**for(j = i+1; j <= max; j++)**

**{**

**recursive(val, p, j, max, new\_total, count);**

**}**

**return;**

**}**

**int nbr\_of\_poss(int val, int p)**

**{**

**int max = sqrt(val);**

**int count = 0;**

**recursive(val, p, 0, max, 0, &count);**

**return count;**

**}**

**int main()**

**{**

**int val, p;**

**scanf("%d%d", &val, &p);**

**printf("%d", nbr\_of\_poss(val, p));**

**return 0;**

**}**

**Counter game**

**#include <stdio.h>**

**#include <string.h>**

**#include <math.h>**

**#include <stdlib.h>**

**int isPow2(long unsigned int);**

**unsigned long int largePow(long unsigned int);**

**int main() {**

**/\* Enter your code here. Read input from STDIN. Print output to STDOUT \*/**

**int t,i,win;**

**long unsigned int n;**

**scanf("%d",&t);**

**for(i=0;i<t;++i)**

**{**

**win=0;**

**scanf("%lu",&n);**

**if(n==1)**

**printf("Richard\n");**

**else**

**{**

**while(n!=1)**

**{**

**if(isPow2(n))**

**n>>=1;**

**else**

**n-=largePow(n);**

**++win;**

**}**

**}**

**if(win%2==0)**

**printf("Richard\n");**

**else**

**printf("Louise\n");**

**}**

**return 0;**

**}**

**int isPow2(long unsigned int n)**

**{**

**return !(n&(n-1));**

**}**

**long unsigned int largePow(long unsigned int n)**

**{**

**long unsigned int m;**

**while(n)**

**{**

**m=n;**

**n=n&(n-1);**

**}**

**return m;**

**}**