#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <assert.h>

#define MAX\_CHARACTERS 1005

#define MAX\_PARAGRAPHS 5

struct word {

char\* data;

};

struct sentence {

struct word\* data;

int word\_count;//denotes number of words in a sentence

};

struct paragraph {

struct sentence\* data ;

int sentence\_count;//denotes number of sentences in a paragraph

};

struct document {

struct paragraph\* data;

int paragraph\_count;//denotes number of paragraphs in a document

};

struct word get\_word(char\* text, int beg, int end) {

struct word answer;

answer.data = calloc(end - beg + 2, sizeof(char));

int index = 0;

int i;

for (i = beg; i <= end; i++)

answer.data[index++] = text[i];

answer.data[index] = 0;

return answer;

}

struct sentence get\_sentence(char\* text, int beg, int end) {

struct sentence answer;

answer.word\_count = 1;

int i;

for (i = beg; i <= end; i++)

if (text[i] == ' ')

++answer.word\_count;

answer.data = calloc(answer.word\_count, sizeof(struct word));

int start = beg;

int index = 0;

for (i = beg; i <= end; i++)

if (text[i] == ' ')

{

answer.data[index++] = get\_word(text, start, i - 1);

start = i + 1;

}

answer.data[index] = get\_word(text, start, i - 1);

return answer;

}

struct paragraph get\_paragraph(char\* text, int beg, int end) {

struct paragraph answer;

answer.sentence\_count = 0;

int i;

for (i = beg; i <= end; i++)

if (text[i] == '.')

++answer.sentence\_count;

answer.data = calloc(answer.sentence\_count, sizeof(struct sentence));

int start = beg;

int index = 0;

for (i = beg; i <= end; i++)

if (text[i] == '.')

{

answer.data[index++] = get\_sentence(text, start, i - 1);

start = i + 1;

}

return answer;

}

struct document get\_document(char\* text) {

struct document answer;

answer.paragraph\_count = 1;

int i;

for (i = 0; text[i]; i++)

if (text[i] == '\n')

++answer.paragraph\_count;

answer.data = calloc(answer.paragraph\_count, sizeof(struct paragraph));

int start = 0;

int index = 0;

for (i = 0; text[i]; i++)

if (text[i] == '\n')

{

answer.data[index++] = get\_paragraph(text, start, i - 1);

start = i + 1;

}

answer.data[index] = get\_paragraph(text, start, i - 1);

return answer;

}

struct word kth\_word\_in\_mth\_sentence\_of\_nth\_paragraph(struct document Doc, int k, int m, int n) {

return Doc.data[n - 1].data[m - 1].data[k - 1];

}

struct sentence kth\_sentence\_in\_mth\_paragraph(struct document Doc, int k, int m) {

return Doc.data[m - 1].data[k - 1];

}

struct paragraph kth\_paragraph(struct document Doc, int k) {

return Doc.data[k - 1];

}

void print\_word(struct word w) {

printf("%s", w.data);

}

int i;

void print\_sentence(struct sentence sen) {

for( i = 0; i < sen.word\_count; i++) {

print\_word(sen.data[i]);

if (i != sen.word\_count - 1) {

printf(" ");

}

}

}

void print\_paragraph(struct paragraph para) {

for( i = 0; i < para.sentence\_count; i++){

print\_sentence(para.data[i]);

printf(".");

}

}

void print\_document(struct document doc) {

for( i = 0; i < doc.paragraph\_count; i++) {

print\_paragraph(doc.data[i]);

if (i != doc.paragraph\_count - 1)

printf("\n");

}

}

char\* get\_input\_text() {

int paragraph\_count;

scanf("%d", &paragraph\_count);

char p[MAX\_PARAGRAPHS][MAX\_CHARACTERS], doc[MAX\_CHARACTERS];

memset(doc, 0, sizeof(doc));

getchar();

for ( i = 0; i < paragraph\_count; i++) {

scanf("%[^\n]%\*c", p[i]);

strcat(doc, p[i]);

if (i != paragraph\_count - 1)

strcat(doc, "\n");

}

char\* returnDoc = (char\*)malloc((strlen (doc)+1) \* (sizeof(char)));

strcpy(returnDoc, doc);

return returnDoc;

}

int main() {

char\* text = get\_input\_text();

struct document Doc = get\_document(text);

int q;

scanf("%d", &q);

while (q--) {

int type;

scanf("%d", &type);

if (type == 3){

int k, m, n;

scanf("%d %d %d", &k, &m, &n);

struct word w = kth\_word\_in\_mth\_sentence\_of\_nth\_paragraph(Doc, k, m, n);

print\_word(w);

}

else if (type == 2) {

int k, m;

scanf("%d %d", &k, &m);

struct sentence sen= kth\_sentence\_in\_mth\_paragraph(Doc, k, m);

print\_sentence(sen);

}

else{

int k;

scanf("%d", &k);

struct paragraph para = kth\_paragraph(Doc, k);

print\_paragraph(para);

}

printf("\n");

}

}