В функции сделать реверс каждого слова в строке

int main()

{int n;

int end = 0, start = 0, counter = 0, i = 0;

char\* string;

printf("Enter a number of elements: \n");

scanf\_s("%d", &n);

if (!(string = (char\*)calloc(n + 1, sizeof(char))))

{return -1;}

rewind(stdin);

printf("Enter a string: \n");

fgets(string, n + 1, stdin);

puts(string);

while (string[i] != '\n' && string[i] != '\0')

{start = i;

while (string[i] != ' ' && string[i] != '\n')

{i++;}

end = i - 1;

counter++;

if (counter % 2 != 0)

{ReverceWord(string, start, end);}

i++;}

puts(string);

free(string);

return 0;}

void ReverceWord(char\* string, int start, int end)

{for (int i = start; i < end; i++)

{char temp = string[i];

string[i] = string[end];

string[end] = temp;

end--; } }

Дан массив строк, отсортировать слова в каждой строке в алфавитном порядке.

int getint(int\* s) {

char temp;

while ((temp = getchar()) != '\n') {

if (temp < '0' temp>'9') {

return 0;

break; } }

return 1;}

char\* enter(int\* str\_len) {

\*str\_len = 1;

char\* str;

str = (char\*)calloc(\*str\_len,1);

while ((str[\*str\_len-1] = getchar()) != '\n') {

(\*str\_len)++;

str = (char\*)realloc(str,(\*str\_len) \* 1);

}

str = (char\*)realloc(str,(\*str\_len + 1) \*1);

str[\*str\_len] = '\0';

return str;

}

void wordCount(char\* str, int\* words) {

for (int i = 0; str[i] != '\0'; i++) {

if (letterCheck(str[i]) == 1 && letterCheck(str[i + 1]) == 0) (\*words)++;

}

}

int letterCheck(char s) {

if ((s >= 65 && s <= 90) (s >= 97 && s <= 122)) return 1;

else return 0;

}

void wordLength(char\* str, int words, int pos,int\*word,int\*wordst,int\*end) {

if (pos > words) {

printf("error.");

return;

}

int w = 0;

int i;

for (i = 0; w!=pos; i++) {

if (letterCheck(str[i]) == 1 && letterCheck(str[i + 1]) == 0) w++;

}

i--;

for (i; letterCheck(str[i]) != 0; i--) (\*word)++;

\*wordst = i+1;

\*end = \*word + \*wordst-1;

}

void reverse(char\* str, int start, int end) {

int mid = (end-start) / 2;

for (int i = 0; i<= mid; i++) {

swap(str, start + i, end - i);

}

Отсортировать строки, для ввода использовать командную строку. Написать свою функцию для сравнения.

int main()

{ int n;

int length = 0, position = 0, lengthOfLastWord = 0, counter = 0;

char\* string;

printf("Enter a number of elements in string: \n");

scanf\_s("%d", &n);

printf("Enter a string: \n");

string = (char\*)calloc(n + 1, sizeof(char));

rewind(stdin);

fgets(string, n + 1, stdin);

for (int i = n; i >= 0; i--)

{while (string[i] != ' ')

{i--;

lengthOfLastWord++;

if (string[i] == ' ')

{counter++; }}

if (counter == 1)

{i--;

while (string[i] != ' ')

{position = i;

length++;

i--;

}}}

DeleteWord(string, length, position, n);

puts(string);

free(string);

return 0;}

void DeleteWord(char\* string, int length, int position, int lengthOfString)

{string[position] = string[position + length + 1];

if (position < lengthOfString)

{position++;

DeleteWord(string, length, position, lengthOfString);}

Создать массив строк(в каждой строке по одной фамилии). Отсортировать строки по алфавиту(любая сортировка int main()

{int i = 0, start = 0, length = 0, count = 0;

int n;

char\* string;

printf("Enter a number of elements in string: \n");

scanf\_s("%d", &n);

string = (char\*)calloc(n + 1, sizeof(char));

rewind(stdin);

fgets(string, n + 1, stdin);

puts(string);

while (string[i] != '\n' && string[i] != '\0')

{i++;

length++;}

for (int j = length; j > 0; j--)

{while (string[j] != ' ')

{j--;}if (string[i] = ' ')

{count++;} start = j + 1;

if (count == 1)

{break;}}

ReverseLastWord(string, start, length);

puts(string);

free(string);

return 0;

}

void ReverseLastWord(char\* string, int start, int end)

{char temp;

if (start > end)

{return;}

temp = string[start];

string[start] = string[end];

string[end] = temp;

start++;

end--;

ReverseLastWord(string, start, end);

}