

Example 3

Write a program in c for this problem: we want to encrypt a string using old mobile phones. In old mobile phones, English alphabets were placed on number buttons. For example 'A', 'B' and 'C' was on button for number 1 and ... 'W', 'X', 'Y' and 'Z' on button for number 9 (Find the alphabets on each button yourself). To type 'B', we have to push button 1 two times, or to type 'F' we should push button 2, three times. We want to encrypt each string by converting it to a string in which each character is substituted by the repetition of the button which should be pushed, for example 'F' should be replaced by 222, and then a - is added after each character. Also we use 0 for space. For example "hi i am alireza" should be encrypted to: "44-4444-0-444-0-2-6-0-2-555-444-777-33-9999-2". write the functions for encryption and decryption of the sentences.



Answer: Certainly! Here's a C program that includes functions for encrypting and decrypting sentences based on the old mobile phone button layout you described:

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

char* encryptSentence(const char* sentence) {
    int i, j, count;
    char encrypted[1000];
    int buttonValues[26] = {2, 22, 222, 3, 33, 333, 4, 44, 444, 5, 55, 555, 6, 66, 666, 7, 77, 777, 7777, 8, 88, 888, 9, 99, 999, 9999};

    for (i = 0, j = 0; sentence[i] != '\0'; i++) {
        if (sentence[i] == ' ') {
            encrypted[j++] = '0';
        } else {
            char c = sentence[i];
```

```

        if (c >= 'A' && c <= 'Z')
            c -= 'A';
        else if (c >= 'a' && c <= 'z')
            c -= 'a';
        else
            continue;

        count = buttonValues[c];
        while (count > 0) {
            encrypted[j++] = count % 10 + '0';
            count /= 10;
        }
    }
    if (sentence[i + 1] != ' ')
        encrypted[j++] = '-';
}

encrypted[j] = '\0';
return strdup(encrypted);
}

char* decryptSentence(const char* encrypted) {
    int i, j;
    char decrypted[1000];
    int buttonValues[8] = {0, 11, 222, 3333, 4444, 55555, 666666, 7777777};

    for (i = 0, j = 0; encrypted[i] != '\0'; i++) {
        if (encrypted[i] == '0') {
            decrypted[j++] = ' ';
        } else if (encrypted[i] == '-') {
            continue;
        } else {
            int count = 0;
            while (encrypted[i] != '-' && encrypted[i] != '\0') {
                count++;
                i++;
            }
            i--;

            decrypted[j++] = buttonValues[count];
        }
    }
}

```

```

    decrypted[j] = '\0';
    return strdup(decrypted);
}

int main() {
    const char* originalSentence = "hi i am alireza";

    char* encryptedSentence = encryptSentence(originalSentence);
    printf("Encrypted sentence: %s\n", encryptedSentence);

    char* decryptedSentence = decryptSentence(encryptedSentence);
    printf("Decrypted sentence: %s\n", decryptedSentence);

    free(encryptedSentence);
    free(decryptedSentence);

    return 0;
}

```

Question: The decryption function is wrong.

Answer: Apologies for the mistake in the decryption function. Here's the corrected version:

```

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

char* encryptSentence(const char* sentence) {
    int i, j, count;
    char encrypted[1000];
    int buttonValues[26] = {2, 22, 222, 3, 33, 333, 4, 44, 444, 5, 55, 555, 6, 66, 666, 7, 77, 777, 7777, 8, 88, 888, 9, 99, 999, 9999};

    for (i = 0, j = 0; sentence[i] != '\0'; i++) {
        if (sentence[i] == ' ') {
            encrypted[j++] = '\0';
        } else {
            char c = sentence[i];
            if (c >= 'A' && c <= 'Z')
                c -= 'A';

```

```

        else if (c >= 'a' && c <= 'z')
            c -= 'a';
        else
            continue;

        count = buttonValues[c];
        while (count > 0) {
            encrypted[j++] = count % 10 + '0';
            count /= 10;
        }
    }
    if (sentence[i + 1] != ' ')
        encrypted[j++] = '-';
}

encrypted[j] = '\0';
return strdup(encrypted);
}

char* decryptSentence(const char* encrypted) {
    int i, j;
    char decrypted[1000];
    int buttonValues[10] = {0, 1, 22, 333, 4444, 55555, 666666, 7777777, 88888888, 999999999};

    for (i = 0, j = 0; encrypted[i] != '\0'; i++) {
        if (encrypted[i] == '0') {
            decrypted[j++] = ' ';
        } else if (encrypted[i] == '-') {
            continue;
        } else {
            int count = 0;
            while (encrypted[i] != '-' && encrypted[i] != '\0') {
                count = count * 10 + (encrypted[i] - '0');
                i++;
            }
            i--;

            decrypted[j++] = buttonValues[count];
        }
    }

    decrypted[j] = '\0';
}

```

```

    return strdup(decrypted);
}

int main() {
    const char* originalSentence = "hi i am alireza";

    char* encryptedSentence = encryptSentence(originalSentence);
    printf("Encrypted sentence: %s\n", encryptedSentence);

    char* decryptedSentence = decryptSentence(encryptedSentence);
    printf("Decrypted sentence: %s\n", decryptedSentence);

    free(encryptedSentence);
    free(decryptedSentence);

    return 0;
}

```

Question: The decryption function is wrong.

Answer: Apologies for the mistake in the decryption function. Here's the corrected version:

```

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

char* encryptSentence(const char* sentence) {
    int i, j, count;
    char encrypted[1000];
    char buttonChars[10] = {'0', '1', '2', '3', '4', '5', '6', '7', '8', '9'};
    char buttonValues[26][4] = {
        "2", // A
        "22", // B
        "222", // C
        "3", // D
        "33", // E
        "333", // F
        "4", // G
        "44", // H
        "444", // I
        "5", // J
    }
}

```

```

"55", // K
"555", // L
"6", // M
"66", // N
"666", // O
"7", // P
"77", // Q
"777", // R
"7777", // S
"8", // T
"88", // U
"888", // V
"9", // W
"99", // X
"999", // Y
"9999" // Z
};

for (i = 0, j = 0; sentence[i] != '\0'; i++) {
    char c = sentence[i];
    if (c == ' ') {
        encrypted[j++] = '0';
    } else if (c >= 'A' && c <= 'Z') {
        int index = c - 'A';
        count = strlen(buttonValues[index]);
        strcpy(&encrypted[j], buttonValues[index]);
        j += count;
    } else if (c >= 'a' && c <= 'z') {
        int index = c - 'a';
        count = strlen(buttonValues[index]);
        strcpy(&encrypted[j], buttonValues[index]);
        j += count;
    }
    if (sentence[i] != ' ' && sentence[i + 1] != '\0' && sentence[i + 1] != ' ')
        encrypted[j++] = '-';
}

encrypted[j] = '\0';
return strdup(encrypted);
}

char* decryptSentence(const char* encrypted) {

```

```

int i, j, k;
char decrypted[1000];
char buttonChars[10] = {'0', '1', '2', '3', '4', '5', '6', '7', '8', '9'};
char buttonValues[26][4] = {
    "2", // A
    "22", // B
    "222", // C
    "3", // D
    "33", // E
    "333", // F
    "4", // G
    "44", // H
    "444", // I
    "5", // J
    "55", // K
    "555", // L
    "6", // M
    "66", // N
    "666", // O
    "7", // P
    "77", // Q
    "777", // R
    "7777", // S
    "8", // T
    "88", // U
    "888", // V
    "9", // W
    "99", // X
    "999", // Y
    "9999" // Z
};

```

```

for (i = 0, j = 0; encrypted[i] != '\0'; i++) {
    char c = encrypted[i];
    if (c == '0') {
        decrypted[j++] = ' ';
    } else if (c == '-') {
        if (encrypted[i + 1] == '-')
            decrypted[j++] = '-';
        continue;
    } else {
        for (k = 0; k < 26; k++) {

```

```

        if (strcmp(buttonValues[k], &encrypted[i]) == 0) {
            decrypted[j++] = 'A' + k;
            i += strlen(buttonValues[k]) - 1;
            break;
        }
    }
}

decrypted[j] = '\0';
return strdup(decrypted);
}

int main() {
    const char* originalSentence = "hi i am alireza";

    char* encryptedSentence = encryptSentence(originalSentence);
    printf("Encrypted sentence: %s\n", encryptedSentence);

    char* decryptedSentence = decryptSentence(encryptedSentence);
    printf("Decrypted sentence: %s\n", decryptedSentence);

    free(encryptedSentence);
    free(decryptedSentence);

    return 0;
}

```