



Bonusaufgabe 5

Algorithmen und Programmierung

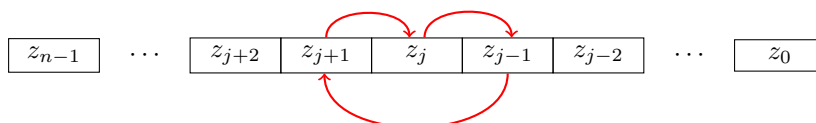
Matthias Werner

Allgemeine Regeln

- Bonusaufgaben sind fakultativ, außer für Studierende des Bachelors Informatik und Kommunikationswissenschaften (B_CC), hier sind sie Vorleistung.
- Die Lösung soll ausschließlich über OpenSubmit abgegeben werden: <https://osg.informatik.tu-chemnitz.de/submit/>
- Bis zum Abgabende (Deadline) können beliebig neue Lösungen abgegeben werden, die die jeweils älteren Versionen ersetzen.
- Eine Lösung wird nur bewertet, wenn der Eingangstest von OpenSubmit positiv ausfällt. Das Ergebnis ist im Status Ihrer Abgabe im OpenSubmit dargestellt; Sie werden außerdem bei einem negativen Ergebnis per E-Mail informiert. Bei negativen Eingangstest können Sie (bis zum Abgabende) jederzeit eine neue Lösung einreichen.
- Ihr Programm muss auf der Testmaschine übersetzbar und lauffähig sein. Die exakten Daten der Testmaschine und des Compilers erfahren Sie über den Link „Test Machines“ im OpenSubmit-Dashboard.
- Ihre Abgabe besteht aus einem ZIP- oder TAR-Archiv, welches ausschließlich ihre Quelldatei enthält. Insbesondere sollte das Archiv kein Unterverzeichnis enthalten.
- Zur Übersetzung werden folgende Optionen genutzt:
`-std=c99 -Wall -Wextra -Wpedantic -Werror -c`

Für diese Aufgabe benutzen wir den rechtseitigen **Dreieringtausch** der Stellen einer positiven ganzen Zahl. Dabei tauschen einige Ziffern einer Zahl ihren Platz. Die Zahl z habe n Ziffern, z_{n-1}, \dots, z_0 , d.h. eine Ziffer z_i hat in der Zahl den Wert $z_i \cdot 10^i$. Beim Ringtausch um die Stelle j tauschen die Ziffern z_{j+1} , z_j und z_{j-1} auf folgende Weise ihren Platz:

- Die Ziffer z_{j+1} kommt an die Stelle j .
- Die Ziffer z_j kommt an die Stelle $j - 1$.
- Die Ziffer z_{j-1} kommt an die Stelle $j + 1$.



- Bei Dreieringtauschoperationen für Zahlen mit weniger als 3 Stellen werden 3 Stellen (führende Null(en)) angenommen, d.h., für alle $n < 3$ wird $n = 3$ gesetzt.
- Dreieringtauschoperationen um die Stelle j mit $j \leq 0$ oder $j \geq n - 1$ sind nicht definiert und dürfen nicht vorkommen.

Beispiele:

- $z = 123456, j = 4 \Rightarrow 312456$
- $z = 123456, j = 1 \Rightarrow 123645$
- $z = 112233, j = 2 \Rightarrow 113223$
- $z = 100, j = 1 \Rightarrow 010 = 10$
- $z = 1, j = 1 \Rightarrow 100$

Dieser Ringtausch wird in zwei von Ihnen zu lösenden Problemen benutzt. Bitte beachten Sie die für beide Aufgaben geltenden Bedingungen, die nach den Aufgabenstellungen folgen.

Problem 1 (4 Punkte)

Schreiben Sie eine C-Funktion `unsigned int maxringshift(unsigned int z, unsigned int k)`, die für eine gegebene Zahl z die größte Zahl zurückgibt, die durch **maximal** k Dreieringtausch-Operationen entstehen kann.

Beispiele

- `maxringshift(1234,2) ~ 3412`
- `maxringshift(1234,1) ~ 3124`
- `maxringshift(4321,2) ~ 4321`
- `maxringshift(213,2) ~ 321`
- `maxringshift(1,2) ~ 100`

Problem 2 (4 Punkte)

Schreiben Sie eine C-Funktion `int *allringshift(unsigned int z, unsigned int k)`, die für eine gegebene Zahl z die einen Zeiger auf ein Array mit allen **unterschiedlichen** Zahlen zurückgibt, die durch **genau** k Dreieringtausch-Operationen entstehen. Das Array soll als Endkennzeichnung als letzten Wert -1 enthalten.

Beispiele

- `allringshift(1234,2) ~ {1342,2143,2314,3412}`
- `allringshift(1234,3) ~ {1234,1342,2314,2431,3241,4132,4213}`
- `allringshift(33333,2) ~ {33333}`
- `allringshift(121,3) ~ {121}`

Beachten Sie bei der Lösung **beider Probleme** folgende **Bedingungen**:

- Beide Funktionen (und evtl. nötige weitere Funktionen) sollen in **einer** Quelldatei enthalten sein.
- Die Datei, die Sie (archiviert, siehe allgemeine Bedingungen) einreichen, **muss** den Namen `ringshift.c` haben.

- Ihr Code wird als Modul übersetzt und darf **keine** main()-Funktion, **keine** globalen Variablen besitzen, und **keine** Ausgabe enthalten. Falls Sie für Debugging-Zwecke main() und/oder Ausgaben benutzen, löschen Sie entsprechende Codeteile vor der Abgabe oder kommentieren Sie diese aus!
- Sie dürfen alle Funktionen der Standardbibliothek nutzen¹, aber **keine weiteren** Bibliotheken.
- Sie dürfen **nicht** die Signaturen der zu definierenden Funktionen **ändern**.
- Ihr Code sollte von beliebigen Hauptfunktionen aus nutzbar sein.

Hinweise:

- Die Übersetzungsoptionen behandeln alle Warnungen als Fehler. Stellen Sie sicher, dass ihr Code keine Warnungen generiert.
- Wenn Sie weitere Funktionen definieren, dürfen diese von beiden Funktionen genutzt werden.
- Die Verwendung ihrer Funktionen könnte z.B. durch folgenden Code erfolgen:

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

extern unsigned int maxringshift(unsigned int z, unsigned int k);
extern int *allringshift(unsigned int z, unsigned int k);
enum {EndOfArray=-1};

int main(int argc, char** argv){
    if (argc!=3) return -1; // wrong # of arguments

    unsigned int z, k;
    if (sscanf(argv[1],"%u",&z) != 1) return -2; // malformed argument
    if (sscanf(argv[2],"%u",&k) != 1) return -2; // malformed argument

    printf("The maximal number after at most %u ringshifts: %u.\n",k,maxringshift(z,k));

    int* res = allringshift(z,k);
    if (res==NULL) return -3; // illegal pointer

    printf("Set of resulting numbers for %u ringshifts: {" ,k);
    for(int i=0; res[i] != EndOfArray; i++){
        if (i!=0) printf(", ");
        printf("%u", res[i]);
    }
    printf("}\n");
    free(res);
    return 0;
}
```

- Sie können die Rohfassung des Codes sowie für Ihre Tests den Quelltext der main()-Funktion auf der Aufgabenseite herunterladen:
<https://osg.informatik.tu-chemnitz.de/lehre/aup/Templates/ringshift.c>
<https://osg.informatik.tu-chemnitz.de/lehre/aup/Templates/ringshift-main.c>
- Sie können bei dieser Aufgabe bis zu acht Punkte erhalten. Entdeckte Betrugsversuche führen zu 0 Punkten.

¹Es gilt die Standardbibliothek in der Konfiguration der Testmaschine von OpenSubmit.