

Grundlagen der Informatik I – 5. Abschnitt :: Aufgaben

Aufgabe 1

Schreiben Sie ein C++-Programm, welches eine Zahl daraufhin untersucht, ob sie negativ, positiv oder Null ist. Implementieren Sie dazu eine Funktion „sign“, welche eine ganze Zahl als Argument erhält und 0 bei Null, 1 bei positiv oder -1 bei negativ zurückgibt:

```
int sign (int z)
```

Innerhalb der `main()`-Funktion soll eine Zahl vom Nutzer gelesen werden und mit Hilfe der Funktion „sign“ eine Aussage über deren Vorzeichencharakter getroffen werden.

Aufgabe 2

Schreiben Sie ein C++ Programm, welches die Quersumme einer 3-stelligen pos. Zahl berechnet. Dazu soll eine Funktion „quersumme3“ implementiert werden, welche als Argument eine ganze Zahl bekommt und als Ergebnis die Quersumme zurückgibt.

```
int quersumme3 (int z)
```

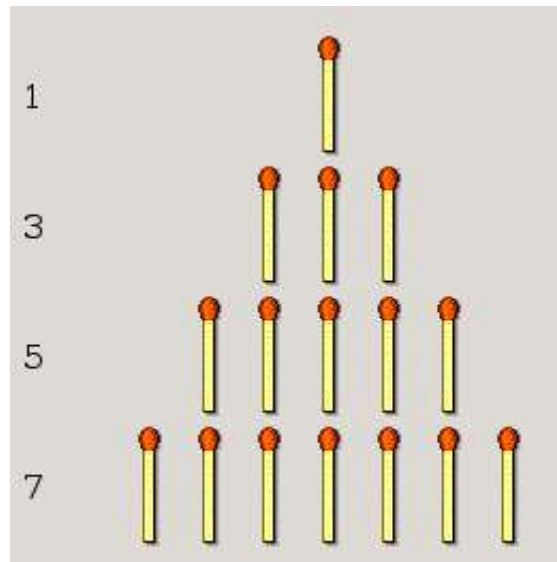
Demonstrieren Sie die Funktionsweise von „quersumme3“ innerhalb der Hauptprogrammfunktion „main()“, indem Sie eine 3-stellige Zahl vom Nutzer einlesen und die Quersumme berechnen.

Hinweis: Es soll davon ausgegangen werden, dass der Nutzer wirklich immer 3-stellige positive Zahlen eingibt.

Zusatzaufgabe für die Hacker unter euch: Stellen Sie sicher, dass die Zahlen immer genau 3 Ziffern lang und positiv sind und geben Sie bei Missachtung eine Fehlermeldung anstatt der Quersumme aus.

Komplexaufgabe – NIM-Spiel

Das NIM-Spiel ist ein Spiel für 2 Spieler, bei dem verschiedene Streichholz-Haufen vorliegen und abwechselnd 1 bis 3 Hölzer von genau einem Haufen entnommen werden. Am Ende gewinnt der Spieler, der die letzten Hölzer insgesamt aufgenommen hat.



Schreiben Sie ein C++-Programm, welches ein solches Spiel für 2 Spieler implementiert.

Implementieren Sie zur Anzeige der Streichholzhaufen eine Funktion „show_heap“, welche 4 Argumente bekommt (Anzahl Hölzer pro Haufen) und dann dem Nutzer anschaulich darstellt:

```
void show_heap(int h1, int h2, int h3, int h4)
```

Zur Anzeige eines Streichholzes kann das Zeichen „|“ dienen. (AltGr + '>').

Stellen Sie sicher, dass auch stets eine gültige Eingabe gemacht wurde:

- die Anzahl der gezogenen Hölzer darf nur 1 bis 3 sein
- der Haufen von dem gezogen wird, darf nur 1 bis 4 sein
- es dürfen nicht mehr Hölzer entfernt werden, als vorhanden sind

Testen Sie das Spiel mit einem Nachbarn Ihrer Wahl und nutzen Sie die restliche Zeit zur Erforschung geeigneter Spielstrategien! ;-)

Zusatzaufgabe für die ganz Harten:

Ersetzen Sie den 2. Spieler durch einen Computergegner, der zufällig von einem Haufen 1 bis 4 zufällig 1 bis 3 Streichhölzer entfernt.

Der Zufallsgenerator wird in „main()“ initialisiert mit: „srand(time(NULL));“

Eine Zufallszahl von 1 bis 3 erhält man mit: „zufall = (rand() % 3) + 1;“

Eine Zufallszahl von 1 bis 4 erhält man mit: „zufall = (rand() % 4) + 1;“

Beispielablauf:

```
1: |  
3: | | |  
5: | | | | |  
7: | | | | | | |
```

Spieler 1 ist am Zug:

Von welchem Haufen Streichhoelzer entfernen? (1-4): 3
Wieviele Hoelzer entfernen?: 3

```
1: |  
3: | | |  
2: | |  
7: | | | | | | |
```

Spieler 2 ist am Zug:

Von welchem Haufen Streichhoelzer entfernen? (1-4): 4
Wieviele Hoelzer entfernen?: 3

```
1: |  
3: | | |  
2: | |  
4: | | | |
```

... und so weiter ...

```
1: |  
0:  
1: |  
0:
```

Spieler 2 ist am Zug:

Von welchem Haufen Streichhoelzer entfernen? (1-4): 1
Wieviele Hoelzer entfernen?: 1

```
0:  
0:  
1: |  
0:
```

Spieler 1 ist am Zug:

Von welchem Haufen Streichhoelzer entfernen? (1-4): 3
Wieviele Hoelzer entfernen?: 1

Spieler 1 gewinnt!

Aufgabe 3

Aufgabe:

- Implementierung der Funktion `gdiv` in C++
- Entwicklung eines kleinen Testrahmens, der die Überprüfung der korrekten Arbeitsweise der Funktion `gdiv` gestattet

Spezifikation der Funktion `gdiv`:

Semantik:

Die ganzzahlige Division zweier Zahlen soll mittels Subtraktion durchgeführt werden. Es soll davon ausgegangen werden, dass nur positive Zahlen eingegeben werden.

Parameter:

- Divisor
- Dividend

Rückgabewert:

- Quotient (ganzzahlig)

Aufgabe 4

Aufgabe:

- Implementierung der Funktion `fulldiv` in C++
- Entwicklung eines kleinen Testrahmens, der die Überprüfung der korrekten Arbeitsweise der Funktion `fulldiv` gestattet

Spezifikation der Funktion `fulldiv`:

Semantik:

Die ganzzahlige Division zweier Zahlen soll mittels Subtraktion wie in Aufgabe 3 durchgeführt werden. Jetzt soll aber zusätzlich noch der Rest zurückgegeben werden.

Parameter:

- Divisor (Call By Value)
- Dividend (Call By Value)
- Rückgabeparameter für Ganzzahlteil (Call By Reference)
- Rückgabeparameter für Restteil (Call By Reference)

Rückgabewert:

- keiner

Zusatzaufgabe:

Schreiben Sie eine Funktion `fulldivsave`, welche einen logischen Rückgabewert besitzt, welcher „false“ ist, falls der Dividend 0 ist und „true“ falls nicht und testen Sie die Arbeitsweise der Funktion `fulldivsave` in einem kleinen Testrahmen.