

Welche Aussagen sind wahr

- ☒ A In einem Array werden mehrere Objekte gleichen Typs zusammengefasst
- ☒ B Die einzelnen Feldelemente werden durch Indizes angesprochen
- ☒ C Ein Array kann unterschiedliche Werte beinhalten
- ☐ D Die Werte in einem Array sind immer konstant

Welche Aussagen sind wahr

- ☒ A Ein Array kann unterschiedliche Werte beinhalten
- ☐ B Die Werte in einem Array sind immer konstant
- ☒ C In einem Array werden mehrere Objekte gleichen Typs zusammengefasst
- ☒ D Die einzelnen Feldelemente werden durch Indizes angesprochen

```
#include<iostream.h>
int main() {
    int u = 1;
    char e[] = "lefid";
    while (u != 3) {
        switch (u){
            case 0: u=3; break;
            case 1: u=1;
            case 3: u=4; break;
            case 4: u=2; break;
            default: u=3;
        }
        cout << e[u];
    }
}
```

Notieren Sie die Ausgabe.

dfi

```
#include<iostream.h>
int main() {
    int u = 1;
    char e[] = "rwefd";
    while (u != 2) {
        switch (u){
            case 0: u=2; break;
            case 1: u=4;
            case 2: u=3; break;
            case 4: u=1; break;
            default: u=0;
        }
        cout << e[u];
    }
}
```

Notieren Sie die Ausgabe.

fre

```
#include<iostream.h>
int main() {
    int a[] = {1,2,0,1};
    int b[] = {4,1,3,3};
    cout << a[b[2]];
    cout << b[a[0]];
}
```

Notieren Sie die Ausgabe.

11

```
#include<iostream.h>
int main() {
    int a[] = {4,2,0,1};
    int b[] = {4,1,4,3};
    cout << a[b[3]];
    cout << b[a[1]];
}
```

Notieren Sie die Ausgabe.

14

```
i[t] = e[p];
```

Welche Aussagen sind wahr:

A t ist ein Array.

~~B~~ e kann ein Array vom Typ `double` sein

C Die Länge des Feldes i stimmt mit der Länge des Feldes e nie überein.

D Einem Element des Feldes e wird ein Wert zugewiesen.

```
e[p] = r[og];
```

Welche Aussagen sind wahr:

~~A~~ r ist ein Array.

~~B~~ Die Länge des Feldes r kann mit der Länge des Feldes e übereinstimmen.

~~C~~ e kann ein Array vom Typ `float` sein

~~D~~ Einem Element des Feldes e wird ein Wert zugewiesen.

```
#include<iostream.h>
int main() {
    int c = 0;
    double f[3]={1,0,2};
    while (f[c] > 0) {
        switch (c) {
            case 1: cout << 'f';
                    c = 0;
                    break;
            case 0: cout << '*';
                    c = 2;
                    break;
            default: cout<< 'd';
                    c = 1;
        }
    }
    return 0;
}
```

Notieren Sie die Ausgabe.

*d

```
#include<iostream.h>
int main() {
    int c = 0;
    double f[4]={1,0,2,3};
    while (f[c] != 2) {
        switch (c) {
            case 3: cout << 'e';
                    c = 2;
                    break;
            case 1: cout << 'v';
                    c = 3;
                    break;
            default: cout<< 'f';
                    c = 1;
        }
    }
    return 0;
}
```

Notieren Sie die Ausgabe.

fve

Gegeben Sei die folgende
Hashfunktion $h(k) = k \bmod 10$
Ergibt das Hashing für $k = 3$ und
 $k = 13$ eine Kollision?

~~A~~ ja

B nein

Gegeben Sei die folgende
Hashfunktion $h(k) = k \bmod 5$
Ergibt das Hashing für $k = 2$ und
 $k = 13$ eine Kollision?

A ja

~~B~~ nein

Die Elemente eines Feldes (eines Arrays)

A sind immer der Grösse nach sortiert

~~B~~ werden über ihren Index angesprochen

C sind immer ganzzahlig

D werden bei jeder Verwendung des Array neu sortiert.

Die Elemente eines Feldes (eines Arrays)

~~A~~ können der Grösse nach sortiert sein

B werden über ihren Wert angesprochen

C werden bei jeder Verwendung des Array neu sortiert.

D sind immer ganzzahlig

```
#include<iostream.h>
int main() {
    int c = 0;
    char f[]="itep";
    while (f[(c+1)]!= '\0'){
        cout << f[c];
        c = c + 1;
    }
    return 0;
}
```

Notieren Sie die Ausgabe

ite

```
#include<iostream.h>
int main() {
    int c = 1;
    char f[]="peti";
    while (f[c] != '\0'){
        cout << f[(c-1)];
        c = c + 1;
    }
    return 0;
}
```

Notieren Sie die Ausgabe

pet

Welche Aussagen sind wahr?

- A Eine Technik zur Datenorganisation ist der B⁺ Zweig
- B Beim Hashing werden vom Programmierer bewußtseins-erweiternde Substanzen eingenommen.
- ☒ Hashing kann eine Kollision verursachen
- D Sollte die Hashfunktion für zwei Elemente denselben Wert ergeben, wird der Benutzer immer gefragt welches Element angezeigt werden soll.

Welche Aussagen sind wahr?

- ☒ A Eine Möglichkeit zur Kollisionsbehandlung beim Hashing ist double hashing
- ☒ B Eine Technik zur Datenorganisation sind Vektoren
- ☒ C Hashing dient zum sehr schnellen Finden von Elementen.
- ☒ D Sollte die Hashfunktion für zwei Elemente denselben Wert ergeben, wird eine Kollisionsbehandlung ausgelöst.

```
#include<iostream.h>
int main() {
    int e = 0;
    int i[3]={1,3,4};
    char f[]="herde";
    for (; e<=2; e=e+1){
        cout << f[i[e]];
    }
    return 0;
}
```

Notieren Sie die Ausgabe

ede

```
#include<iostream.h>
int main() {
    int e = 2;
    int i[3]={1,3,4};
    char f[]="herde";
    for (; e>=0; e=e-1){
        cout << f[i[e]];
    }
    return 0;
}
```

Notieren Sie die Ausgabe

ede