

Status	Beendet
Begonnen	Mittwoch, 26. Februar 2025, 10:01
Abgeschlossen	Mittwoch, 26. Februar 2025, 11:04
Dauer	1 Stunde 2 Minuten
Bewertung	26,20 von 60,00 (43,67%)

Frage 1

Richtig

Erreichte Punkte 5,00 von 5,00

Ein Spieler gibt drei ganze, positive Zahlen a , b und c ein. Das Ergebnis berechnet sich wie folgt: Ist die Summe der Zahlen gerade, wird die Summe zurückgeliefert. Ist die Summe ungerade, wird das Produkt der drei Zahlen zurückgegeben. Sind jedoch alle Zahlen gleich, beträgt das Ergebnis 1000 unabhängig von der Summe.

Schreiben Sie eine Funktion `int numberGame(int a, int b, int c)`, die das Ergebnis des Spiels berechnet. Sollte mindestens ein Argument negativ sein, soll 0 zurückgeliefert werden.

Zum Beispiel:

Test	Resultat
int a = -2, b = 2, c = 1; printf("Number %d, %d, %d: %d\n", a, b, c, numberGame(a, b, c));	Number -2, 2, 1: 0
int a = 2, b = 1, c = 3; printf("Number %d, %d, %d: %d\n", a, b, c, numberGame(a, b, c));	Number 2, 1, 3: 6
int a = 1, b = 1, c = 1; printf("Number %d, %d, %d: %d\n", a, b, c, numberGame(a, b, c));	Number 1, 1, 1: 1000

Antwort: (Abzugssystem: 0 %)

Antwort zurücksetzen

```

1 int numberGame(int a, int b, int c) {
2     if(a < 0 || b < 0 || c < 0){
3         return 0;
4     }
5     if(a==b && a==c && b==c){
6         return 1000;
7     }
8     else if((a+b+c) % 2 == 0){
9         return a+b+c;
10    }
11    return a*b*c;
12 }

```

	Test	Erwartet	Erhalten	
✓	int a = 2, b = 1, c = 3; printf("Number %d, %d, %d: %d\n", a, b, c, numberGame(a, b, c));	Number 2, 1, 3: 6	Number 2, 1, 3: 6	✓
✓	int a = 1, b = 3, c = 1; printf("Number %d, %d, %d: %d\n", a, b, c, numberGame(a, b, c));	Number 1, 3, 1: 3	Number 1, 3, 1: 3	✓
✓	int a = 1, b = 1, c = 1; printf("Number %d, %d, %d: %d\n", a, b, c, numberGame(a, b, c));	Number 1, 1, 1: 1000	Number 1, 1, 1: 1000	✓

Alle Tests bestanden! ✓

Richtig

Bewertung für diese Einreichung: 5,00/5,00.

Frage 2

Richtig

Erreichte Punkte 5,00 von 5,00

Gegeben sei das folgende Feld. Wie sehen die Inhalte von `arr` nach der Abarbeitung folgender Anweisungen aus? Tragen Sie die Werte unten ein!

```
int arr[] = {3, 1, 0, 2, 7, 1}, *p = arr;  
p += 3;  
*p = 4;  
arr[0] = *arr - *p;  
arr[2] += *(arr + 4);  
(*++p)--;
```

`arr[0]:` ✓

`arr[1]:` ✓

`arr[2]:` ✓

`arr[3]:` ✓

`arr[4]:` ✓

`arr[5]:` ✓

Frage 3

Falsch

Erreichte Punkte 0,00 von 8,00

Schreiben Sie eine Funktion `float* berechneZeilensummen (int n, int m, float mat[][m])`, die eine zeilenweise als zweidimensionales Array abgespeicherte $n \times m$ -Matrix `mat` übergeben bekommt, für jede Zeile die Summe berechnet und ein (eindimensionales) **dynamisch allozierten** Array mit den Zeilensummen der Matrix zurückliefert.

Zum Beispiel:

Test	Resultat
<pre>float m3[][3] = {{2, 31, 7}, {1, 48, 3}, {7, 8, 9}}; float *d = berechneZeilensummen(3, 3, m3); printf("%.1f, %.1f, %.1f\n", d[0], d[1], d[2]); free(d);</pre>	(40.0, 52.0, 24.0)

Antwort: (Abzugssystem: 0 %)

Antwort zurücksetzen

```
1 float* berechneZeilensummen(int n, int m, float mat[][m]) {
2
3     if(n > 0 && m > 0){
4         float *arr = (float*)malloc(n * sizeof(float));
5         if(arr){
6             for(int i = 0; i<n; i++){
7                 for(int j = 0; j<m; j++){
8                     arr[i] += mat[i][j];
9                 }
10            }
11        }
12        return arr;
13    }
14    return NULL;
15 }
```

	Test	Erwartet	Erhalten	
✓	<pre>float m3[][3] = {{2, 31, 7}, {1, 48, 3}, {7, 8, 9}}; float *d = berechneZeilensummen(3, 3, m3); printf("%.1f, %.1f, %.1f\n", d[0], d[1], d[2]); free(d);</pre>	(40.0, 52.0, 24.0)	(40.0, 52.0, 24.0)	✓
✗	<pre>float m4[][4] = {{0, -2, 0, 2}, {2, 0, 0, 4}, {0, 0, 2, 6}, {0, 0, 0, 1}}; float *d = berechneZeilensummen(4, 4, m4); printf("%.1f, %.1f, %.1f, %.1f\n", d[0], d[1], d[2], d[3]); free(d);</pre>	(0.0, 6.0, 8.0, 1.0)	(1058905907447988224.0, 6.0, 8.0, 1.0)	✗

Einige verborgene Testfälle sind ebenfalls fehlgeschlagen.

Ihr Code muss alle Tests bestehen, um eine Bewertung zu erhalten. Versuchen Sie es noch einmal.

Unterschiede anzeigen

Falsch

Bewertung für diese Einreichung: 0,00/8,00.

Frage 4

Falsch

Erreichte Punkte 0,00 von 1,00

Gegeben ist die folgende (zugegebenermaßen wenig sinnvolle) Struktur zur Verwaltung von Studierenden. Tragen Sie für die Anmeldung an Index 123 für das allererste Modul die Note 1,3 ein. (Achtung, wegen der automatischen Auswertung bitte keine unnötigen Leerzeichen eingeben.)

```
typedef struct {
    char name[80];
    unsigned int matr_nr;
    struct {
        char name[80];
        float note;
    } module[40];
} Student;
Student anmeldungen[200];
```

Antwort: anmeldungen[123].Student.module[0].note = 1.3

✗

Frage 5

Teilweise richtig

Erreichte Punkte 1,20 von 2,00

Bitte wählen Sie im folgenden Lückentext die passenden Anweisungen aus, um die Summe zweier Zahlen durch (Pseudo-) Call-by-Reference zu erzeugen. Die Summe soll danach in die Variable *a* geschrieben werden.

```
void ✓ berechneSumme( int *a ✓ , int *b ✗ )
{
    a ✗ += b ✓ ;
}
```

```
int main() {
    int c = 5, d = 3;
    berechneSumme(&c, d);
    printf("%d", c);
}
```

Die Antwort ist teilweise richtig.

Sie haben 3 richtig ausgewählt.

Frage 6

Richtig

Erreichte Punkte 4,00 von 4,00

Überführen Sie den folgenden Text in ein C-Struct.

Es soll eine Person beschrieben werden. Jede "Person" besitzt folgende Eigenschaften:

- einen **Namen**, der maximal 40 Zeichen lang ist
- ein **Alter**, welches durch eine speichersparende, nicht-negative Ganzzahl dargestellt wird
- eine **Adresse**, die durch eine eigene Datenstruktur repräsentiert wird. Diese enthält eine 30-stellige **Strasse**, einen **Ort** aus maximal 50 Stellen und eine **PLZ**.

Für die Lösung benutzen Sie die Bezeichnungen aus dem Text. Verwenden Sie nur *Kleinbuchstaben* für die einzelnen Strukturkomponenten!

```
struct Person {  
    char name[40];  
    unsigned short alter;  
    struct adr {  
        char strasse[30];  
        char ort[50];  
        unsigned int plz;  
    } adresse;  
} p;
```

Frage 7

Falsch

Erreichte Punkte 0,00 von 4,00

Was berechnet folgende Funktion?

Tip: Führen Sie ggf. dazu (handschriftlich oder am Rechner) einen sog. Programmtrace durch, d.h. spielen Sie den Algorithmus mit Testdaten durch und dokumentieren Sie die Werte innerhalb der Schleifendurchläufe.

```
unsigned int what_do_i_do (unsigned int n) {  
    unsigned int r = 0;  
    while (n) {  
        r = r * 10 + (n % 10);  
        n /= 10;  
    }  
    return r;  
}
```

- ☒ a. Nichts sinnvolles, weil versehentlich r statt n zurückgegeben wird ✖
- ☐ b. Binärdarstellung von n
- ☐ c. Gar nichts, die Funktion bleibt in einer Endlosschleife hängen
- ☐ d. Umgekehrte Ziffernfolge von n
- ☐ e. Anzahl aller ungeraden Zahlen von 1 bis n
- ☐ f. Alternierende Quersumme von n
- ☐ g. Iterierte Quersumme von n

Die Antwort ist falsch.

Frage 8

Richtig

Erreichte Punkte 1,00 von 1,00

Sie möchten mit Hilfe einer Präprozessordirektive die Konstante "SEC_PER_MIN" definieren, die die Anzahl der Sekunden pro Minute enthält. Geben Sie die entsprechende Codezeile an.

Antwort:



Frage 9

Richtig

Erreichte Punkte 2,00 von 2,00

Variablen, die mit den ✓ Datentypen int, long, float usw. deklariert wurden, können in arithmetischen Ausdrücken verwendet werden.

+, -, *, /, % sind arithmetische ✓ .

Geben Sie in den beiden folgenden, **je nacheinander ausgeführten** Programmzeilen den jeweiligen Wert an.

```
int n=5, m=4;
```

```
n += m;    // n hat jetzt den Wert  ✓
```

```
n /= m;    // n hat nun den Wert  ✓
```

Mit einem Casting von int auf ✓ kann auch der reelle Anteil dargestellt werden. Wie lautet die Divisionsanweisung mit entsprechendem Casting?

```
float o =  ✓ n/m;
```


Frage 10

Falsch

Erreichte Punkte 0,00 von 6,00

Schreiben Sie eine **rekursive** Funktion, die eine Zeichenkette rückwärts ausgibt.

Hinweis: Für eine nicht-rekursive Lösung gibt es **keine** Punkte, auch wenn alle Tests bestanden wurden (manuelle Überprüfung)!

Zum Beispiel:

Test	Resultat
char *str = "Test"; printRev(str);	tseT
char *str = "c programming is cool"; printRev(str);	looc si gnimmargorp c

Antwort: (Abzugssystem: 0 %)

[Antwort zurücksetzen](#)

```

1 void printRev (char *str) {
2     int len = strlen(str);
3     if(len == 0){
4         return;
5     }
6
7     printf("%c", *(str+len-1));
8     //printRev(*(str+len-1));
9 }
```

	Test	Erwartet	Erhalten	
✗	char *str = "Test"; printRev(str);	tseT	t	✗
✗	char *str = "Noch ein Test!"; printRev(str);	!tseT nie hcoN	!	✗
✗	char *str = "c programming is cool"; printRev(str);	looc si gnimmargorp c	l	✗
✓	char *str = ""; printRev(str);			✓

Einige verborgene Testfälle sind ebenfalls fehlgeschlagen.

Ihr Code muss alle Tests bestehen, um eine Bewertung zu erhalten. Versuchen Sie es noch einmal.

[Unterschiede anzeigen](#)

Falsch

Bewertung für diese Einreichung: 0,00/6,00.

Frage 11

Falsch

Erreichte Punkte 0,00 von 8,00

Schreiben Sie eine Funktion `int* extGerade (int arr[], int in_n, int *out_n)`, die für ein als Parameter übergebenes Feld `arr` mit `in_n` Ganzzahlen ein neues, dynamisch alloziertes Array der richtigen Länge zurückliefert, das **alle geraden** Werte aus dem Ursprungsfeld enthält. Falls **keine gerade Zahl** vorhanden ist, soll `out_n` auf 0 gesetzt werden und es darf kein neues Array auf dem Heap angelegt werden.

Damit die aufrufende Funktion auch die Länge des dynamisch erzeugten Feldes kennt, ist der Parameter `out_n` auf die passende Länge zu setzen bzw. auf 0, falls kein Speicher angefordert wurde (oder keiner alloziert werden konnte).

Antwort: (Abzugssystem: 0 %)

Antwort zurücksetzen

```

1 int* extGerade(int arr[], int in_n, int *out_n) {
2     *out_n = 0;
3     if(in_n < 0){
4         for(int i = 0; i < in_n; i++){
5             if(arr[i]%2 == 0){
6                 out_n++;
7             }
8         }
9     }
10    if(*out_n > 0){
11        int *newArr = (int*)malloc(*out_n * sizeof(int));
12        if (newArr) {
13            for (int i = 0; i < *out_n; i++) {
14                for(int j = 0; j < in_n; j++){
15                    if(arr[j] % 2 == 0){
16                        newArr[i] = arr[j];
17                    }
18                }
19            }
20        }
21        return newArr;
22    }
23    }
24
25    return NULL;
26
27 }

```

	Test	Erwartet	
✗	<pre> int arr[] = {1, 2, 9, 6, 3}; int neu_n, *neua = extGerade(arr, 5, &neu_n); if (neua) { for (int i=0; i<neu_n; i++) printf("%d, ", neua[i]); free(neua); } </pre>	2, 6,	✗
✗	<pre> int arr[] = {1, 7, 9822}; int neu_n, *neua = extGerade(arr, 3, &neu_n); if (neua) { for (int i=0; i<neu_n; i++) printf("%d, ", neua[i]); free(neua); } </pre>	9822,	✗

Einige verborgene Testfälle sind ebenfalls fehlgeschlagen.

Ihr Code muss alle Tests bestehen, um eine Bewertung zu erhalten. Versuchen Sie es noch einmal.

Falsch

Bewertung für diese Einreichung: 0,00/8,00.

Frage 12

Richtig

Erreichte Punkte 1,00 von 1,00

Ordnen Sie zu!

Inhaltsoperator ✓Adressoperator ✓

Die Antwort ist richtig.

Frage 13

Richtig

Erreichte Punkte 2,00 von 2,00

Unten sind Vorschläge für die Deklaration einer Funktion, der zwei Strings übergeben werden und die den zweiten String an den ersten anhängt. Dabei soll kein neuer String erstellt, sondern der erste verändert werden. Welche der Deklarationen ist korrekt und passend für den gedachten Funktionsaufruf?

- ☐ a. char* verknuepfeStrings (char s1, char s2);
- ☐ b. void verknuepfeStrings (char *s1, char s2);
- ☐ c. void verknuepfeStrings (char s1, char s2);
- ☐ d. char* verknuepfeStrings(char *s1, char *s2);
- ☒ e. void verknuepfeStrings(char *s1, char *s2); ✓

Die Antwort ist richtig.

Frage 14

Richtig

Erreichte Punkte 1,00 von 1,00

Was ist die Ausgabe?

```
char str[] = "Always look on the bright side of life.";
printf("%c\n", str[10]);
```

- ☐ a. Leerzeichen
- ☐ b. Fehler
- ☐ c. o
- ☐ d. n
- ☒ e. k ✓

Die Antwort ist richtig.

Frage 15

Richtig

Erreichte Punkte 1,00 von 1,00

Welches Schlüsselwort wird in C verwendet, um eine Variable zu deklarieren, deren Wert nach der Initialisierung nicht mehr verändert werden kann?

Antwort: **Frage 16**

Richtig

Erreichte Punkte 3,00 von 3,00

Ein vorzeichenloses 8-Bit Speicherwort a sei mit $(0101\ 0011)_2$ vorbelegt. Welche der folgenden Aussagen sind richtig?

- ☒ a. Durch einen Rechtsshift wird der Wert durch 2 geteilt. ✓
- ☒ b. $a \gg 2$ ergibt 0001 0100 ✓
- ☒ c. $a \& 0000\ 0001$ ergibt 0000 0001 ✓
- ☐ d. Der Wert von a ist gerade.
- ☒ e. a hat den Wert 83. ✓
- ☐ f. $a \mid 0000\ 0001$ ergibt 0000 0001
- ☐ g. a hat den Wert 53

Die Antwort ist richtig.

Frage 17

Falsch

Erreichte Punkte 0,00 von 6,00

Implementieren Sie die Funktion `Vector skalarMultiplikation(const Vector *a, float s)`, die einen Vektor, welcher als folgende Struktur definiert ist, mit einem Skalar multipliziert:

```
typedef struct {  
    float x, y, z;  
} Vector;
```

Zum Beispiel:

Test	Resultat
<pre>Vector v1 = {2.0, 2.0, 1.0}; float s = 2.1; Vector res = skalarMultiplikation(&v1, s); printf("Skalarmultiplikation von v1 und s: (%.4f, %.4f, %.4f)\n", res.x, res.y, res.z);</pre>	Skalarmultiplikation von v1 und s: (4.2000, 4.2000, 2.1000)

Antwort: (Abzugssystem: 0 %)

Antwort zurücksetzen

```
1 | Vector skalarMultiplikation(const Vector *a, float s) {  
2 |  
3 | }
```

Syntaxfehler

```
__tester__.c: In function 'skalarMultiplikation':  
__tester__.c:14:1: error: control reaches end of non-void function [-Werror=return-type]  
14 | }  
   | ^  
cc1: all warnings being treated as errors
```

Falsch

Bewertung für diese Einreichung: 0,00/6,00.

[← Nachrichten und Ankündigungen](#)

Direkt zu: