

Hochschule Deggendorf	Prüfungsteilnehmer
Prüfungsfach: Einführung in die Programmierung	Platzziffer
Aufgabensteller: Prof. Dr. P. Jüttner	
Prüfungstermin: 7.2.11	
10 Aufgaben, Arbeitszeit: 90 Min.	
zugelassene Hilfsmittel: alle	
☞ Keinen Rotstift verwenden! ☞ nur an den vorgesehenen Stellen antworten! ☞ erst denken, dann die Lösung hinschreiben	☞ ggf. Rückseite oder Zusatzblatt beschreiben! ☞ Viel Erfolg

Datentypen

1. Welche Werte haben die Variablen uc, c1, f1 am Ende des folgenden Programms (jeweils als Zahlen interpretiert)? (5 Punkte, empfohlene Bearbeitungszeit 5min)

```

void main(void)
{ unsigned char uc;
  char c1;
  c1 = -3;
  unsigned long l = 272;
  float f1 = 6.2 + 9/4;
  uc = c1;
  c1 = l;
}

```

Shiftoperationen

2. Welchen Wert haben die Variablen i1, i2, i3 am Ende des folgenden Programms? (Überlegen Sie bei i3 genau!) (4 Punkte, empfohlene Bearbeitungszeit 4 min)

```
void main(void)
{ int i1, i2, i3, i4;

  i1 = 1;
  i2 = 1;
  i3 = -35;
  i1 = i1 >> 2;
  i2 = i2 << 4;
  i3 = i3 >> 1;

}
```

Bitweise Operatoren

3. Schreiben Sie ein C-Programm, das das zweite Bit einer einzulesenden Zahl vom Typ short auf 1 setzt, falls das letzte Bit (16., niederwertigstes Bit) gleich 0 ist. Alle anderen Bits sollen unverändert bleiben. Falls das letzte Bit gesetzt ist (also gleich 1), soll keine Aktion durchgeführt werden. Danach soll die Zahl auf Bildschirm ausgegeben werden. #include-Anweisungen dürfen weggelassen werden.

Beispiel: 0000111000011111 wird unverändert ausgegeben, aus 0011111111111110 soll 0111111111111110 werden. (12 Punkte, empfohlene Bearbeitungszeit 12 min)

Kontrollstrukturen

4. Gegeben sind die folgenden Programmstücke

```
int i = 1;
int j = 3;
while(i<10)
{ j = j*};
  i++;
}
```

```
int j = 3;
for (int i =0; i<=10; i=i++)
{ j = j*};
}
```

Warum entspricht die for-Schleife nicht der while-Schleife? Was muss in der for-Schleife geändert werden, dass beide Programmstücke äquivalent sind (d.h. dass j nach Ausführung identische Werte hat? (6 Punkte, empfohlene Bearbeitungszeit 6 min)

Vektoren, Zeichenketten, Prozeduren und Funktionen

5. Schreiben Sie eine C-Prozedur, die in einer Zeichenkette alle kleinen Buchstaben a und b „verdoppelt“. Dabei werden aus einem kleinen ‘a’ (bzw. ‘b’) in der Zeichenkette zwei nacheinander folgende ‘a’s (‘aa’) bzw. ‘b’s (‘bb’). Z.B. Aus der Zeichenkette „Hans ist besonders schlau“ wird „Haans ist bbesonders schlaau“.
- Die Prozedur soll zwei Parameter besitzen, einen in dem die originale Zeichenkette an die Prozedur übergeben wird (diese bleibt unverändert), einen Parameter für die veränderte Zeichenkette. (z.B. void doppel_a_b(char orig[], char neu[])
- Zum Feststellen der Länge der originalen Zeichenkette kann entweder auf das Endezeichen ‘\0’ abgefragt werden oder die Funktion strlen(orig) aufgerufen werden. Gehen Sie davon aus, dass die originale Zeichenkette richtig mit einem Wert besetzt ist und der Parameter für die veränderte Zeichenkette diese aufnehmen kann, Fehlerabfragen müssen nicht implementiert werden.
- (15 Punkte, empfohlene Bearbeitungszeit 15min)

6. Schreiben Sie ein C-Funktion, die für zwei Mengen von Zahlen $M1 = \{1, 3, 5, 7, 9\}$ und $M2 = \{-3, -7, 5, 0, 6\}$ das Maximum des Absolutbetrags aller möglichen Differenzen von jeweils einer Zahl aus $M1$ und einer Zahl aus $M2$ berechnet und als Funktionswert zurückgibt.
- Zur Berechnung des Absolutbetrags dürfen Sie davon ausgehen, dass eine Funktion `unsigned int abs(int z)`, die den Absolutbetrag einer `int` Zahl als Funktionswert berechnet, bereits definiert ist. Ein Aufruf von `abs` lautet z.B: `i = abs(z);`

Tipp:

- a.) Speichern Sie die Elemente der zwei Mengen in zwei Vektoren und führen Sie die Differenzberechnung mittels `for`-Schleifen durch. Die Vektoren können als lokale Variable in der Funktion definiert werden.
- b.) Zur Berechnung des Absolutbetrags dürfen Sie davon ausgehen, dass eine Funktion `abs`, die den Absolutbetrag einer `int` Zahl als Funktion berechnet bereits definiert ist. Ein Aufruf von `abs` lautet z.B: `i = abs(z);`

(15 Punkte, empfohlene Bearbeitungszeit 15min)

7. Schreiben Sie eine C-Funktion, die so lange Gleitkommazahlen von Tastatur einliest, bis der Wert 0 eingegeben wird. Als Funktionsergebnis soll das Produkt der eingelesenen Zahlen zurückgegeben werden. Wird als erste Zahl sofort 0 eingegeben, so soll das Funktionsergebnis 1 sein.
(9 Punkte, empfohlene Bearbeitungszeit 9 min)

Pointer

8. Welche Werte haben die Variablen a, b und c am Ende (vor der return-Anweisung) des folgenden Programms (interpretiert als int)?
(6 Punkte, empfohlene Bearbeitungszeit 6 min)

```
int main()
{ int a = -3;
  int b = 10;
  int c = -b;;
  int *ip1 = &a;
  int *ip2 = ip1;
  *ip1 = *ip1 * -(*ip2);
  ip2 = &b;
  c = c * -(*ip2);
  ip1 = &b;
  ip2 = &c;
  *ip1 = b + *ip2;
  *ip2 = *ip1 / a;

  return 0;
}
```

9. Schreiben Sie eine C-Funktion, die feststellt, ob zwei Pointer auf int-Zahlen, die per Parameter übergeben werden, auf die gleiche Adresse zeigen. In diesem Fall soll 1 zurückgegeben werden, andernfalls 0.
(6 Punkte, empfohlene Bearbeitungszeit 6 min)

Datenstrukturen**10. Datenstruktur Patient**

Definieren Sie in C eine Datenstruktur zur Erfassung der Patientendaten in einer Arztpraxis. Folgende Daten eines Patienten sollen in getrennten Komponenten der Datenstruktur mit geeigneten Standarddatentypen und ggf. Vektoren gespeichert werden:

- Vorname
- Nachname
- Versicherungsart (Privatpatient / Kassenpatient)
- Krankenkasse

(4 Punkte, empfohlene Bearbeitungszeit 4 min)