Übung vom 01.07.98 bis 07.07.98

Aufgabe 27:

Was leistet das folgende Programm? Warum würde der Versuch, den Wert des Pointers **string** in *main()* zu ändern, zu einer Fehlermeldung des Compilers führen, während dies in *conv()* offensichtlich funktioniert? Wo erfolgt die Definition der Macros *isupper(), tolower()* und *toupper()?* Warum erscheint in dem Befehl *scanf()* vor der Variable **string** kein **&**? Wie können Sie das Programm beenden?

```
#include <stdio.h>
#include <ctype.h>
void conv(char *string)
{
    while(*string)
    {
        if(isupper(*string))
            *string = tolower(*string);
        else
            *string = toupper(*string);
        string++;
    }
}

int main()
{
    char string[128];
    while(1)
    {
        printf("Bitte geben Sie ein Wort ein: ");
        if(scanf("%s", string)<EOF)
            break;
        conv(string);
        printf("Konvertiert: %s\n", string);
    }
    return 0;
}
```

Aufgabe 28:

Angenommen, die Definitionszeile **int i=10, j=1, *p, z[10];** sei in einem Programm enthalten. Was sind dann die Inhalte der jeweils verwendeten Variablen nach folgenden Anweisungen?

a) p = &z[5];
b) p = &j; i = *p;
c) p = z; i = *p++;
d) p = &j; i = (*p)++;
e) p = z; i = *++p;
f) p = &z[7]; p++; i = p-&z[4];

Was ist an den folgenden Anweisungen falsch:

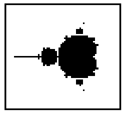
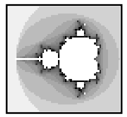
g) z[10] = 10;
h) *z+1 = 1;
i) p = &z[5]; *(p-6) = 4;

Die Mandelbrotmenge

Die Mandelbrotmenge – benannt nach Benoit Mandelbrot – ist eine Menge von komplexen Zahlen. Die Zugehörigkeit einer komplexen Zahl c zu dieser Menge kann mittels eines Tests, basierend auf der Formel $z = z^2 + c$, bestimmt werden, wobei die zu überprüfende Zahl c konstant bleibt und über die komplexe Zahl z iteriert wird (d.h. z wird ständig neu berechnet, beginnend mit $z = (0, 0)$).

Die Zahl c gehört genau dann zur Mandelbrotmenge, wenn der Betrag von z ein gewisses Limit (normalerweise 2) nie überschreitet – also wenn bei jeder Iteration der obigen Formel $|z| < 2$ gilt.

Das linke der beiden folgenden Bilder zeigt die Mandelbrotmenge (schwarz) im Bereich [(-2.2, -1.5), (1.0, 1.5)] der komplexen Zahlenebene. Um ein Bild interessanter zu gestalten färbt man die Bereiche der komplexen Zahlenebene ein, die nicht zur Mandelbrotmenge gehören, wobei für jeden getesteten Punkt der Ebene die Anzahl der benötigten Iterationen bis zum Überschreiten des Limits den Farbwert bestimmt. Im rechten Bild entsprechen dunkle/schwarze Bereiche hohen Iterationstiefen; die Mandelbrotmenge selbst wurde weiß belassen.

Aufgabe 29:(Abgabe bis Mi. 15.07.98, 12 Uhr)

a) Legen Sie in Ihrem Home-Verzeichnis ein Unterverzeichnis (z.B. mandel) an und kopieren Sie die Dateien mandel.c, draw.c und draw.h in dieses Verzeichnis. Sie finden diese Dateien im Verzeichnis /proj/i4cing/mandel/ oder im WWW unter der URL http://www4.informatik.uni-erlangen.de/Lehre/SS98/V_C/Uebung/ . Die Datei mandel.c enthält ein Grundgerüst zum Generieren von "Apfelmännchen" (Bilder der Mandelbrotmenge) und soll von Ihnen um die fehlende Funktionalität erweitert werden. Die beiden anderen Dateien bilden das Modul draw, das zum Speichern der Bilder als Postscript- oder Bitmapdatei gebraucht wird.
b) Erweitern Sie mandel.c um eine Funktion "iterate", die als Argument die zu testende komplexe Zahl c und die maximale Anzahl der durchzuführenden Iterationsschritte erhält. Sie soll so lange die Berechnung $z = z^2 + c$ wiederholen, bis entweder $|z|$ das Limit überschreitet oder die maximale Iterationstiefe erreicht wird. Als Ergebnis liefert die Funktion die Anzahl der durchgeführten Iterationen.

Verwenden Sie für die nötigen Berechnungen die in Aufgabe 26 von Ihnen implementierten Operationen auf komplexen Zahlen.

c) Ein Bild besteht aus vielen Einzelpunkten (Pixel), die Punkten in der komplexen Zahlenebene entsprechen. Schreiben Sie eine Funktion, die in Abhängigkeit vom gewählten Ausschnitt der Komplexen Zahlenebene und den Dimensionen des Bildes (festgelegt durch die Variablen width und height in mandel.c) einem Pixel die ihm entsprechende komplexe Zahl zuordnet.

d) Eine weitere Funktion soll die für das Bild gewünschte Zuordung der Iterationstiefe zu Farbwerten vornehmen. In den Tafelübungen wird eine einfache Funktion vorgestellt, die auch in mandel.c teilweise vorgegeben ist. Um schönere Bilder, evtl. auch in Farbe, zu erzeugen, sind hier der Phantasie keine Grenzen gesetzt...

e) Schreiben Sie die noch fehlende Funktion compute_image (ihr Prototyp ist bereits in mandel.c vorgegeben). Sie muß für jeden Bildpunkt (Pixel) mit der Funktion draw_set_color aus dem draw-Modul einen Farbwert setzen. Um diesen zu erhalten, muß zuerst die dem Pixel zugeordnete komplexe Zahl ermittelt werden (siehe Teilaufgabe c). Dann wird für diese Zahl die Iterationstiefe bestimmt (siehe Teilaufgabe b) und der entsprechende Farbwert gewählt (siehe Teilaufgabe d).
Bem: Die Variable relevant_iter legt fest, welche Iterationstiefen für die Farbgebung interessant sind, das dies i.d.R. wesentlich weniger als die maximale Iterationstiefe sind.

f) Die Bearbeitung dieser Teilaufgabe ist bis auf die Eingabe des Dateinames freiwillig.
In der Main-Funktion sind bisher alle Werte fest kodiert und können bei der Programmausführung nicht vom Benutzer eingegeben werden. Es wäre besser, wenn der Benutzer die jeweiligen Parameter bei der Programmausfüuhrung eingeben könnte. Insbesondere soll der fest vorgegebene Dateiname "test" durch einen vom Benutzer festzulegenden Namen

Bemerkungen zur Programmieraufgabe:

Kompilieren:

- auf HP (CIP-Pool, 1. Stock):
  cc -Ae -o mandel mandel.c complex.c draw.c -lm
- auf Sun (CIP-Pool, !. Stock, Nebenraum):
  cc -Aa -o mandel mandel.c complex.c draw.c -lm
- Linux:
  In draw.c Kommentare der Zeilen 9 und 10 entfernen (INTEL_STUFF/LINUX_STUFF)
  gcc -o mandel mandel.c complex.c draw.c -lm
- Windows/NT
  In draw.c Kommentar der Zeile 9 entfernen (INTEL_STUFF)
  Projekt erstellen und Kompilieren, je nach verwendetem Kompiler...
  Leider kann ich hier nicht viel weiterhelfen...

Das Programm erzeugt eine Datei mit dem angegebenen Dateinamen und der Endung ".ps" falls draw_write_psverwendet wird, und mit ".bmp", falls draw_write_bmp verwendet wird. Diese Dateien enthalten das berechnete Bild und können mit dem Programm xv angezeigt werden (z.B. xv test.ps). Das xv-Programm wird beendet, indem die Taste 'q' im entsprechenden Fenster gedrückt wird. Auf Windows-Rechnern kann das mitgelieferte Zeichenprogramm zum Betrachten einer Bitmapdatei (.bmp) verwendet werden.

Klausur:

am Do. 30.07.1998 um 8:00

in der Mensa