8.1.A Sonstiges/Variable Argumentlisten – Arbeitsblatt

Aufgabe 1: Produkt

Die Funktion `multiprod()` soll das Produkt beliebig vieler Zahlen vom Typ `double` zurückgeben. Hier ein Beispiel:

```
1  double x;
2  x=multiprod(3, 5.0, 2.0, 3.0);
```

Hier soll der Aufruf das Ergebnis der Rechenoperation $5 \cdot 2 \cdot 3 = 30$ zurückgeben. Der erste Parameter ist die Anzahl der Faktoren. Er soll fest sein, die übrigen optional. Wenn man die Funktion mit nur einem Parameter `0` aufruft, soll das Ergebnis `1` betragen.

Aufgabe 2: Differenz

Die Funktion `multidiff()` soll von einer Zahl beliebig viele andere Zahlen vom Typ `double` abziehen. Hier ein Beispiel:

```
1  double x;
2  x=multidiff(4, 3.5, 0.1, 0.2, 0.3, 0.4);
```

Hier soll der Aufruf das Ergebnis der Rechenoperation $3,5-0,1-0,2-0,3-0,4 = 2,5$ zurückgeben. Der erste Parameter ist die Anzahl der Subtrahenden, der zweite ist der Minuend. Die ersten beiden Parameter sollen fest sein, die übrigen optional.

Aufgabe 3: Strings leeren

Die Funktion `loeschestrings()` soll beliebig viele Strings ausleeren. Das kann dadurch geschehen, dass das Zeichen Nr. 0 des jeweiligen Strings durch den Terminator `'\0'` ersetzt wird. Das Ende der Parameterliste soll daran zu erkennen sein, dass ein Nullzeiger übergeben wird. Hier ein Beispiel:

```
1  char s[80]="Hund";
2  char t[80]="Katze";
3  char u[80]="Maus";
4  loeschestrings(s, t, u, NULL);
```

Aufgabe 4: Beliebig viele Strings anhängen

Die Funktion `strmulticat()` soll an einen String `s` beliebig viele weitere Strings anhängen. Dieser soll der erste Parameter sein. Der zweite Parameter soll die maximale Länge dieses Strings sein.

```
1  char s[81]="Und er sagte: ";
2  strmulticat(s, 80, /* siehe unten */);
```

Was die Ende-Erkennung der Parameterliste angeht, so sind mehrere Varianten zu realisieren:

a) Der dritte Parameter ist die Anzahl der Anhänge:

```
1  strmulticat1(s, 80, 3, "Hallo", "du", "da");
```

b) Der dritte Parameter ist die Anzahl der maximal anzuhängenden Bytes:

```
1  strmulticat2(s, 80, 9, "Hallo", "du", "da");
```

c) Es wird so lange angehängt, bis ein Leerstring erkannt wird:

```
1    strmulticat3(s, 80, "Hallo", "du", "da", "");
```

d) Es wird so lange angehängt, bis ein Anhang dieselbe Adresse wie `s` selbst hat:

```
1    strmulticat4(s, 80, "Hallo", "du", "da", s);
```

e) Es wird so lange angehängt, bis ein Anhang einem im dritten Parameter vereinbarten Schlüsselwort gleicht:

```
1    strmulticat5(s, 80, "ENDE", "Hallo", "du", "da", "ENDE");
```

f) Es wird so lange angehängt, bis ein Nullzeiger erkannt wird:

```
1    strmulticat6(s, 80, "Hallo", "du", "da", NULL);
```