

1^ο Ερώτημα

Πρώτη ρουτίνα

Τελεστές	Αριθμός Εμφανίσεων	Έντελα	Αριθμός
Void	1	count	4
sort_numbers_ascending()	1	temp	3
int	3	i	5
number[]	8	j	8
{}	4	k	7
,	5	0	2
;	6	1	1
for	3	Numbers in ascending order:\n	1
=	6	%d\n	1
<	3	n2=9	N2=32
"++"	3		
>	1		
+	1		
if	1		
printf()	2		
n1=15	N1=48		

Παραδοχή: Στα for loops δεν έχει συμπεριληφθεί το «;» σαν επιπλέον τελεστής αφού ανήκει στο for.

Δεύτερη ρουτίνα

Τελεστές	Αριθμός Εμφανίσεων	Εντελα	Αριθμός
main()	1	count	5
{}	2	number	1
int	1	20	2
number[]	2	0	2
,	7	"How many numbers you are going to enter:"	1
;	8	"\nEnter the numbers one by one:"	1
=	2	t	2
printf()	3	%d	3
while()	1	\nThis is a test	1
>	1	i	5
<	1	n2=10	N2=23
scanf()	3		
&	3		
for()	1		
"++"	1		
void	1		
sort_numbers_ascending()	1		
n1=17	N1=39		

Τρίτη ρουτίνα

Τελεστές	Αριθμός Εμφανίσεων	Εντελα	Αριθμός
void()	1	i	16
main()	1	20	4
{}	6	0	4
int	2	n	3
num[]	9	count	7
,	13	j	7
;	16	a	4
printf()	6	x	1
&	5	b	1
while()	1	How many numbers you are going to enter:	1
<	5	%d ή %d/n	6
>	2	\nEnter the numbers one by one:	1
"_"	1	t	5
"++"	4	\nThis is a test	1
+	1	\nThis is my test	1
if()	1	Numbers in ascending order:\n	1
=	9	1	1
scanf()	5	n2=17	N2=64
for()	5		
n1=19	N1=93		

Παραδοχή: Εφόσον το &d και το %d/n ανήκουν σε string type της printf, τα θεωρώ ως ιδιά.

2^ο Ερώτημα

Ο τύπος του Halstead για την εύρεση του μήκος προγράμματος σύμφωνα με την θεωρία είναι ως εξής:

$$N_{est} = n_1 \log_2 n_1 + n_2 \log_2 n_2$$

$$N = N_1 + N_2$$

Για την εύρεση του επιπέδου προγράμματος του Halstead (L), Θα χρειαστούμε το V^* όπου αντιστοιχεί στον όγκο της πιο συνοπτικής λύσης κάθε ρουτίνας.

$$V = N \log_2 n$$

$$n = n_1 + n_2$$

$$L = V^* / V$$

$$\lambda = L V^*$$

1^η ρουτίνα:

```
Nest=87.1326839471086
N=80
Nest/N=1.0891585493388576
V*=V= 366.79700005769257, αφού αποτελεί ήδη ο κώδικας την πιο σύντομη λύση.
L=1
Λ=366.79700005769257
LOComm/PLOCode=1/20
```

```
1  /* Fuction for getting sorting number in ascending order*/
2  void sort_numbers_ascending(int number[], int count)
3  {
4      int temp, i, j, k;
5      for (j = 0; j < count; ++j)
6      {
7          for (k = j + 1; k < count; ++k)
8          {
9              if (number[j] > number[k])
10             {
11                 temp = number[j];
12                 number[j] = number[k];
13                 number[k] = temp;
14             }
15         }
16     }
17     printf("Numbers in ascending order:\n");
18     for (i = 0; i < count; ++i)
19         printf("%d\n", number[i]);
20 }
```

2^η ρουτίνα:

```
Nest=102.70614925012941
N=62
Nest/N= 1.656550794356926
V=294.8030251341351
V*= 220.07820003461552
L=0.746
λ= 164,178
LOComm/PLOCode=1/16
```

Παρατηρούμε πως το t παραμένει πάντα 0 οπότε δεν μπαίνει ποτέ ο κώδικας στο while. Άρα μπορούμε να αφαιρέσουμε τα παρακάτω από τον κώδικα:

```
t=0
,
while (t>20)
{
printf("\nThis is a test");
scanf("%d", &count);
}
```

Μετά την αφαίρεση των παραπάνω τελεστών και εντελών προκύπτει:

```
n1=16
N1=32
n2=8
N2=16
N=48
n=24
```

```
1 void main()
2 {
3     int i, count, number[20], t=0;
4     printf("How many numbers you are going to enter:");
5     scanf("%d", &count);
6     printf("\nEnter the numbers one by one:");
7     while (t>20)
8     {
9         printf("\nThis is a test");
10        scanf("%d", &count);
11    }
12    for (i = 0; i < count; ++i)
13        scanf("%d", &number[i]);
14    /* Calling the Function*/
15    sort_numbers_ascending(number, count);
16 }
```

3^η ρουτίνα

```
Nest= 150.1974910566839
N=157
Nest/N= 0.9566719175584962
V=737.9690357481516
V*=524.6619474737779
L= 0,710
λ=373,009
LOComm/PLOCode=12/45
```

Παρατηρούμε πως το t παραμένει πάντα 0 οπότε δεν μπαίνει ποτέ ο κώδικας στο while και στο for. Επίσης το x και το b αρχικοποιούνται χωρίς να έχουν κάποια χρήση στον κώδικα. Άρα μπορούμε να αφαιρέσουμε τα παρακάτω από τον κώδικα:

```
, t=0

while (t>20)
{
printf("\nThis is a test");
scanf("%d", &count);
printf("\nThis is my test");
scanf("%d", &count);
}

for(t=20; t<20; t--)
{
scanf("%d", &count);
}

, x, b
```

Μετά την αφαίρεση των παραπάνω τελεστών και εντελών προκύπτει:

```
n1=17
N1=63
n2=12
N2=45
N=108
n=29
```

```

1  void main()
2  {
3  int i, num[20], t=0;
4  int n, count, j, a, x, b;
5  printf("How many numbers you are going to enter:");
6  scanf("%d", &count);
7  printf("\nEnter the numbers one by one:");
8  /*
9  *
10 *
11 *
12 Test this code
13 *
14 *
15 *
16 */
17 while (t>20)
18 {
19 /*test*/
20 printf("\nThis is a test");
21 scanf("%d", &count);
22 printf("\nThis is my test");
23 scanf("%d", &count);
24 }
25 for(t=20; t<20; t--)
26 {
27 scanf("%d", &count);
28 }
29 /*My loop begins*/
30 for (i = 0; i < count; ++i)
31 scanf("%d", &num[i]);
32 for (i = 0; i < n; ++i){
33 for (j = i + 1; j < n; ++j){
34 if (num[i] > num[j]){
35 a = num[i];
36 num[i] = num[j];
37 num[j] = a;
38 }
39 }
40 }
41 /*Here are the data*/
42 printf("Numbers in ascending order:\n");
43 for (i = 0; i < count; ++i)
44 printf("%d\n", num[i]);
45 }

```

3^ο Ερώτημα

Σ1.

1^η ρουτίνα

Nest/N=1.0891585493388576

V*=V= 366.79700005769257, αφού αποτελεί ήδη ο κώδικας την πιο σύντομη λύση.

L=1

Λ=366.79700005769257

LOComm/PLOCode=1/20

2^η ρουτίνα

Nest/N= 1.656550794356926

V=294.8030251341351

V*= 220.07820003461552

L=0.746

λ= 164,178

LOComm/PLOCode=1/16

AVERAGE(NEST/N)= (1.089+ 1.656)/2 = 1,372

AVERAGE(V) = (366.797+294.803)/2 = 330.8

AVERAGE(V*)= (366.72+220.078)/2 = 293.399

AVERAGE(L) = (1+0.746)/2 = 0.873

AVERAGE(λ) = (366.797+164,178)/2 = 265.4875

AVERAGE(LOC/PLOC) =1/20 +1/16=0.1125

Σ2.

1^η ρουτίνα

Nest/N₁=1.0891585493388576

V₁*=V₁= 366.79700005769257, αφού αποτελεί ήδη ο κώδικας την πιο σύντομη λύση.

L₁=1

λ₁=366.79700005769257

LOComm₁/PLOCode₁=1/20

N₁=80

2^η ρουτίνα

Nest/N₂= 1.656550794356926

V₂=294.8030251341351

V*₂= 220.07820003461552

L₂=0.746

λ₂= 164.178

LOComm₂/PLOCode₂=1/16

N₂=62

N₁+N₂=142

Weighted average:

WAverage(Nest/N) = (Nest/N₁* N₁ + Nest/N₂ * N₂) / (N₁ + N₂) =1.336

WAverage(L) = (L₁*N₁ + L₂*N₂) / (N₁ + N₂)= 0.889

WAverage(λ) = (λ₁ * N₁ + λ₂ * N₂) / (N₁ + N₂)= 278.329

WAverage(LOComm/PLOCode)= (LOComm1/PLOCode1)*N₁+ (LOComm2/PLOCode2)*N₂)/(N₁+N₂)=
0.055

Σε γενικές περιπτώσεις όπου ο κώδικας έχει αρκετές ρουτίνες είναι προτιμότερο να χρησιμοποιηθεί Weighted Average αφού γίνεται πιο πραγματική μέτρηση.

4^ο Ερώτημα

Στην Β υλοποίηση παρατηρούμε πως υπάρχει μόνο μια ρουτίνα οπότε ο σταθμισμένος μέσος όρος (Weighted Average) θα είναι ίσος με τον Average με βάση το N

```
Nest/N= 0.9566719175584962
V=737.9690357481516
V*=524.6619474737779
L= 0,710
λ=373,009
LOComm/PLOCode=12/45
```

Για Weighted Average 1^{ης} και 2^{ης} ρουτίνας:

Nest/N₁ < Nest/N₂: Λογικό αφού το N₂ είναι μεγαλύτερο

L₁ > L₂: Σωστό αφού η 2^η ρουτίνα δεν είναι υλοποιημένη με τον ελάχιστο απαιτούμενο κώδικα

λ₁ > λ₂: Λογικό αφού εξαρτάται από το L και το V και εξαρτάται από την γλώσσα προγραμματισμού

LOComm/PLOCode₁ < LOComm₂/PLOCode₂: Υπάρχει μεγαλύτερο ποσοστό σχολίων στην 2^η ρουτίνα