

Διασφάλιση Ποιότητας Συστημάτων Λογισμικού

Kristi Cami 3882

Δίνονται οι λειτουργικές προδιαγραφές ενός λογισμικού:

- Κατασκευάστε ένα πρόγραμμα το οποίο να δέχεται ως είσοδο έναν δεκαεξαδικό αριθμό και να επιστρέφει τον αντίστοιχο δεκαδικό αριθμό.
- Το πρόγραμμα θα δέχεται ως είσοδο έναν θετικό δεκαεξαδικό αριθμό με το πολύ 30 ψηφία. Επίσης, επιτρέπονται μόνο κεφαλαίοι χαρακτήρες. Αν δοθούν από το χρήστη περισσότερα ψηφία, τότε αυτά αγνοούνται και λαμβάνονται υπόψη μόνο τα 30 πρώτα. Δεν χρειάζεται να ενημερωθεί ο χρήστης για αυτό. Θα πρέπει όμως στην εμφάνιση του αποτελέσματος να αναγράφεται ο δεκαεξαδικός αριθμός για τον οποίο έγινε η μετατροπή.
- Θα πρέπει να εμφανίζονται μηνύματα λάθους για τις περιπτώσεις που δοθεί κενή είσοδος ή μη έγκυρο δεκαεξαδικό ψηφίο, και να ζητείται εκ νέου είσοδος από τον χρήστη.
- Το πρόγραμμα θα πρέπει να παράγει σωστό αποτέλεσμα στα όρια του `double`, αλλιώς να εμφανίζει σχετικό μήνυμα στο χρήστη.

Ζητούμενα:

1. Σχεδιάστε τις περιπτώσεις ελέγχου που πρέπει να διεξαχθούν με βάση την τεχνική διαμέρισης σε κλάσεις ισοδυναμίας. [20%]
2. Σχεδιάστε τις περιπτώσεις ελέγχου που πρέπει να διεξαχθούν με βάση την τεχνική ανάλυσης οριακών τιμών. [20%]

Απάντηση:

Οι κανόνες που χρησιμοποιούνται παρακάτω βρίσκονται στην διαφάνειες «12.Ποιότητα λογισμικού - Εγκυροποίηση λογισμικού (Μέρος 2).pdf» στην σελίδα 4 για την διαμέριση σε κλάσεις ισοδυναμίας και 14 για Ανάλυση οριακών τιμών .

Διαμέριση σε κλάσεις ισοδυναμίας

Αλλαγές τιμών εισόδου που επηρεάζουν την έξοδο

1. δεκαεξαδικός ≥ 0
2. $0 < \text{ψηφία_δεκαεξαδικού} \leq 30$
3. $(\text{'0'} \leq \text{κάθε_ψηφιο_δεκαεξαδικού} \leq \text{'9'} \mid \mid \text{'A'} \leq \text{κάθε_ψηφιο_δεκαεξαδικού} \leq \text{'F'})$

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

<https://upload.wikimedia.org/wikipedia/commons/5/5c/ASCII-Table-wide.pdf>

4. DBL_MIN<=δεκαεξαδικός<=DBL_MAX

Type	Storage size	Value range	Precision
float	4 byte	1.2E-38 to 3.4E+38	6 decimal places
double	8 byte	2.3E-308 to 1.7E+308	15 decimal places
long double	10 byte	3.4E-4932 to 1.1E+4932	19 decimal places

https://www.tutorialspoint.com/cprogramming/c_data_types.htm

Κλάσεις ισοδυναμίας & περιπτώσεις ελέγχου

Περίπτωση 1 (δεκαεξαδικός>=0)

- Από Κανόνα 2 χρειαζομαι δυο κλάσεις ισοδυναμίας: δεκαεξαδικός>=0 και δεκαεξαδικός<0
- Κλάση δεκαεξαδικός>=0 => (Είσοδος = 7FD9, Έξοδος = 32729)
- Κλάση δεκαεξαδικός<0 => (Είσοδος = -8A20, Έξοδος = Άκυρη τιμή εισόδου, ο αριθμός εισόδου είναι αρνητικός)

$$\begin{aligned}*(-8A20)_{16} &= -[(8 \times 16^3) + (10 \times 16^2) + (2 \times 16^1) + (0 \times 16^0)] = (-35360)_{10} *(7FD9)_{16} &= (7 \times 16^3) + (15 \times 16^2) + (13 \times 16^1) + (9 \times 16^0) = (32729)_{10}\end{aligned}$$

Περίπτωση 2 ($0 < \text{ψηφιά_δεκαεξαδικού} \leq 30$)

- Από Κανόνα 2 χρειαζομαι τρεις κλάσεις ισοδυναμίας: ψηφιά_ δεκαεξαδικού < 0 , $0 \leq \text{ψηφιά_δεκαεξαδικού} \leq 30$ και ψηφιά_ δεκαεξαδικού > 30
- Κλάση ψηφιά_ δεκαεξαδικού $< 0 \Rightarrow$ (Είσοδος = “ “ , Έξοδος = Άκυρη τιμή εισόδου)
- Κλάση , $0 \leq \text{ψηφιά_δεκαεξαδικού} \leq 30 \Rightarrow$ (Είσοδος = 7FD97FD9 , Έξοδος = 2144960473)
- Κλάση ψηφιά_ δεκαεξαδικού $> 30 \Rightarrow$ (Είσοδος = 74AFBB03F6A04FE374AFBB03F6A04FAB , Έξοδος = Ο δεκαεξαδικός αριθμός 74AFBB03F6A04FE374AFBB03F6A04F μετατράπηκε στον δεκαδικό 605870674163677048074051446992445519)

$$*(7FD97FD9)_{16} = (7 \times 16^7) + (15 \times 16^6) + (13 \times 16^5) + (9 \times 16^4) + (7 \times 16^3) + (15 \times 16^2) + (13 \times 16^1) + (9 \times 16^0) = (2144960473)_{10}$$

Περίπτωση 3 (‘0’ \leq κάθε ψηφιο δεκαεξαδικού \leq ‘9’ || ‘A’ \leq κάθε ψηφιο δεκαεξαδικού \leq ‘F’)

- Πρόκειται για μεταβλητή που έχει 16 έγκυρες τιμές {0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F }
- Σύμφωνα με τον κανόνα 3, υπάρχουν δύο κλάσεις ισοδυναμίας:

Κλάση κάθε_ψηφιο_ δεκαεξαδικού IN {0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F } \Rightarrow Περιπτώσεις ελέγχου

(κάθε_ψηφιο_ δεκαεξαδικού = 0, έλεγχος = έγκυρη τιμή),
(κάθε_ψηφιο_ δεκαεξαδικού = 1, έλεγχος = έγκυρη τιμή)
(κάθε_ψηφιο_ δεκαεξαδικού = 2, έλεγχος = έγκυρη τιμή)
(κάθε_ψηφιο_ δεκαεξαδικού = 3, έλεγχος = έγκυρη τιμή)
(κάθε_ψηφιο_ δεκαεξαδικού = 4, έλεγχος = έγκυρη τιμή)
(κάθε_ψηφιο_ δεκαεξαδικού = 5, έλεγχος = έγκυρη τιμή)
(κάθε_ψηφιο_ δεκαεξαδικού = 6, έλεγχος = έγκυρη τιμή)
(κάθε_ψηφιο_ δεκαεξαδικού = 7, έλεγχος = έγκυρη τιμή)
(κάθε_ψηφιο_ δεκαεξαδικού = 8, έλεγχος = έγκυρη τιμή)
(κάθε_ψηφιο_ δεκαεξαδικού = 9, έλεγχος = έγκυρη τιμή)
(κάθε_ψηφιο_ δεκαεξαδικού = A, έλεγχος = έγκυρη τιμή)
(κάθε_ψηφιο_ δεκαεξαδικού = B, έλεγχος = έγκυρη τιμή)
(κάθε_ψηφιο_ δεκαεξαδικού = C, έλεγχος = έγκυρη τιμή)
(κάθε_ψηφιο_ δεκαεξαδικού = D, έλεγχος = έγκυρη τιμή)
(κάθε_ψηφιο_ δεκαεξαδικού = E, έλεγχος = έγκυρη τιμή)
(κάθε_ψηφιο_ δεκαεξαδικού = F, έλεγχος = έγκυρη τιμή)

**Κλάση κάθε_ψηφιο_ δεκαεξαδικού NOT IN {0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F } =>
Περιπτώσεις ελέγχου**

(κάθε_ψηφιο_ δεκαεξαδικού = f, έλεγχος = άκυρη τιμή),
(κάθε_ψηφιο_ δεκαεξαδικού = \$, έλεγχος = άκυρη τιμή),
(κάθε_ψηφιο_ δεκαεξαδικού = " ", έλεγχος = άκυρη τιμή)

Περίπτωση 4 (DBL_MIN<=δεκαεξαδικός<=DBL_MAX)

- Κανόνα 2 χρειαζομαι τρεις κλάσεις ισοδυναμίας: δεκαεξαδικός < DBL_MIN, DBL_MIN<=δεκαεξαδικός<=DBL_MAX και δεκαεξαδικός >DBL_MAX
- Κλάση δεκαεξαδικός < DBL_MIN => (Είσοδος = 1e-308, Έξοδος = Άκυρη τιμή εισόδου)
- Κλάση DBL_MIN<=δεκαεξαδικός<=DBL_MAX => (Είσοδος = 1B45 , Έξοδος = 6981)
- Κλάση δεκαεξαδικός >DBL_MAX => (Είσοδος = 1e308, Έξοδος = Άκυρη τιμή εισόδου)

$$*(1B45)_{16} = (1 \times 16^3) + (11 \times 16^2) + (4 \times 16^1) + (5 \times 16^0) = (6981)_{10}$$

Ανάλυση οριακών τιμών

Παρακάτω συνοψίζονται ξανά οι αλλαγές τιμών εισόδου που επηρεάζουν την έξοδο

1. δεκαεξαδικός >= 0
2. 0<ψηφιά_ δεκαεξαδικού<=30
3. ('0'<=κάθε_ψηφιο_ δεκαεξαδικού<='9' || 'A'<=κάθε_ψηφιο_ δεκαεξαδικού<='F')
4. DBL_MIN<=δεκαεξαδικός<=DBL_MAX

Περίπτωση 1 (δεκαεξαδικός>=0)

Ο περιορισμός αφορά ένα πεδίο τιμών σε ανάλυση οριακών τιμών (Κανόνας 2)

- Έγκυρες τιμές: (Είσοδος =1, Έξοδος =1)
- Άκυρες τιμές: (Είσοδος = -1, Έξοδος =άκυρη τιμή)
- Τιμές στα όρια: (Είσοδος = 0, Έξοδος = 0)

Περίπτωση 2 (0<ψηφιά_ δεκαεξαδικού<=30)

Ο περιορισμός αφορά ένα πεδίο τιμών σε ανάλυση οριακών τιμών (Κανόνας 2)

- Έγκυρες τιμές: (Είσοδος =A, Έξοδος =10), (Είσοδος = 23456789ABCDEF123456789ABCDEF, Έξοδος = 11446129963703441085972389523344879)
- Άκυρες τιμές: (Είσοδος = " ", Έξοδος =άκυρη τιμή)
- Τιμές στα όρια: (Είσοδος =23456789ABCDEF123456789ABCDEF2, Έξοδος =183138079419255057375558232373518066)

Περίπτωση 3 ('0'<=κάθε_ψηφιο_ δεκαεξαδικού<='9' || 'A'<=κάθε_ψηφιο_ δεκαεξαδικού<='F')

Ο περιορισμός αφορά ένα σύνολο τιμών σε ανάλυση οριακών τιμών (Κανόνας 1), οι τιμές στα όρια προκύπτουν από τον πίνακα ascii

- **Έγκυρες τιμές:** Αναλύθηκαν στην διαμέριση σε κλάσεις ισοδυναμίας όλες η έγκυρες τιμές
- **Άκυρες τιμές:** (κάθε_ψηφιο_ δεκαεξαδικού = !, έλεγχος = άκυρη τιμή), (κάθε_ψηφιο_ δεκαεξαδικού = w, έλεγχος = άκυρη τιμή), (κάθε_ψηφιο_ δεκαεξαδικού = X, έλεγχος = άκυρη τιμή)
- **Τιμές στα όρια:** (κάθε_ψηφιο_ δεκαεξαδικού = /, έλεγχος = άκυρη τιμή), (κάθε_ψηφιο_ δεκαεξαδικού = :, έλεγχος = άκυρη τιμή), (κάθε_ψηφιο_ δεκαεξαδικού = G, έλεγχος = άκυρη τιμή), , (κάθε_ψηφιο_ δεκαεξαδικού = @, έλεγχος = άκυρη τιμή)

Περίπτωση 4 DBL_MIN<=δεκαεξαδικός<=DBL_MAX

Ο περιορισμός αφορά ένα πεδίο τιμών σε ανάλυση οριακών τιμών (Κανόνας 2). Παρακάτω δίνονται μερικές ενδεικτικές τιμές για τα όρια του double. **Τα παραδείγματα αυτά δεν χρησιμοποιούνται στην πράξη διότι σύμφωνα με της λειτουργικές προδιαγραφές ο μεγαλύτερος δεκαεξαδικός αριθμός έχει μήκος 30 χαρακτήρων.**

- **Έγκυρες τιμές:** (Είσοδος =23456789ABCDEF123456789ABCDEF2, Έξοδος =183138079419255057375558232373518066)
- **Άκυρες τιμές:** (Είσοδος = 1e308, Έξοδος =άκυρη τιμή), (Είσοδος = 1e-308, Έξοδος =άκυρη τιμή)
- **Τιμές στα όρια:** (Είσοδος = 0x1p-1022 , Έξοδος = 2.22507e-308), (Είσοδος = 0x1.fffffffffffffp+1023, Έξοδος = 1.79769e+308),

Οι τιμές στα όρια προκύπτουν από τον παρακάτω κώδικα.

```
#include <stdio.h>
#include <float.h>

int main() {
    double smallest_double = DBL_MIN;
    double max_double = DBL_MAX;

    printf("DBL_MIN in hexadecimal: %a\n", smallest_double);
    printf("DBL_MAX in hexadecimal: %a\n", max_double);

    return 0;
}
```

Λαμβάνοντας υπόψη μόνο τον κώδικα του λογισμικού:

3. Κατασκευάστε τον γράφο ροής τους προγράμματος και υπολογίστε την κυκλωματική πολυπλοκότητα με 3 διαφορετικούς τρόπους. [20%]

4. Καταγράψτε τα βασικά μονοπάτια και σχεδιάστε τις περιπτώσεις ελέγχου με βάση την τεχνική δοκιμής βασικών μονοπατιών εκτέλεσης. [20%]

5. Ομαδοποιήστε τις περιπτώσεις ελέγχου που βρήκατε από την εφαρμογή όλων προηγούμενων τεχνικών. Έπειτα, εντοπίστε συγκεκριμένα λάθη στον κώδικα (bugs) αξιοποιώντας τις περιπτώσεις ελέγχου που σχεδιάσατε. Ξεκινήστε την απάντησή σας καταγράφοντας για κάθε περίπτωση ελέγχου τα αποτελέσματα (δώστε στιγμιότυπο οθόνης / screenshot) σε σύγκριση με τα αναμενόμενα. Προτείνετε συγκεκριμένες διορθώσεις στον κώδικα για τα λάθη που εντοπίσατε. [20%]

Απάντηση:

```
1 { #include <stdio.h>
    #include <stdlib.h>
    #include <math.h>
    #include <string.h>
    #define N 30

    int main()
    {
        char input[N];
        char *hexarr;
        int i,digit,len,validNum;
        double num;
        printf("Metatropeas apo dekaexadiko se dekadiko systima!\n\n");

2 { do {
        printf("Dwste enan dekaexadikos arithmo (mexri %d psifia):",N);
        fgets(input, sizeof input, stdin);
        input[strlen(input)-1]='\0';
        fflush(stdin);

        len = strlen(input);
        validNum = 1;

3 if (len == 0)

4 { {
        printf("Den dwsate kapoia psifia.\n\n");
        int validNum = 0;
    }

5 { else
        for (i=0;i<len;i++)

6 { if (input[i]<'0' || input[i]>'F' || (input[i]>'9' && input[i]<'A'))
7 { {
10 { printf("To %d psifio den einai egkyro!\n\n",i);
    validNum = 0;
11 break;
```

```

12 }
13 } while (!validNum);
14 hexarr = (char *) malloc(len * sizeof(char));
15 for (i=0;i<len;i++)
16     hexarr[i] = input[len-i];
17 num = 0.0;
18 for (i=0;i<len;i++) {
19     if (hexarr[i]<'9')
20         digit = hexarr[i] - '0';
21     else
22         digit = 10 + hexarr[i] - 'A';
23     num += digit*pow(16,i);
24 }
25 printf("Ο Δεκαεξαδικός '%s' αντιστοιχεί στον δεκαδικό '%.0f'.\n\n",input,num);

```

Υπολογισμός κυκλωματικής πολυπλοκότητας

- $V(g) = e - n + 2 = 31 - 23 + 2 = 10$
- $V(g) = p + 1 = 9 + 1 = 10$
- $V(g)$ = οι περιοχές του γράφου 10
- Άρα θα ψάξω για έως $V(g) = 10$ βασικά μονοπάτια

Βρίσκω βασικά μονοπάτια (με κίτρινο η προσθήκη νέας ακμής/ακμών)

- 1-2-3-4-**13-2**-3-5-6-7-8-9-10-11-12-13-2-3-5-6-7-8-9-13-14-15-17-18-23 => M1
- 1-2-**3-5**-6-7-8-9-13-14-15-17-18-23 => M2
- 1-2-3-5-**6-10**-11-12-13-2-3-5-6-7-8-9-13-14-15-17-18-23 => M3
- 1-2-3-5-6-**7-10**-11-12-13-2-3-5-6-7-8-9-13-14-15-17-18-23 => M4
- 1-2-3-5-6-7-**8-12**-13-14-15-17-18-23 => M5
- 1-2-3-5-6-7-8-**9-12**-13-14-15-17-18-23 => M6
- 1-2-**3-4**-13-2-3-5-6-7-8-**9-12**-13-14-15-17-18-23 => M7
- 1-2-3-4-13-2-3-5-6-7-**8-12**-13-14-15-17-18-23 => M8
- 1-2-3-5-6-7-8-9-12-13-14-**15-16**-15-17-18-23 => M9
- 1-2-3-4-13-2-3-5-6-**7-10**-11-12-13-14-15-17-18-23 => M10

Περιπτώσεις ελέγχου για βασικά μονοπάτια

- **M1 =>**
μήνυμα "dwste enan dekaexadiko arithmo (mexri 30 psifia) "
(Είσοδος = " ")
μήνυμα "Den dwsate kapoia psifia"
(Είσοδος = "@ ")
μήνυμα 'To psifio @ den einai egyro'
(Είσοδος = 0 , Εξοδος=0)
μήνυμα 'Ο Dekaxadikos 0 antistoixei ston dekadiko 0.'
M2 => (Είσοδος = "1A " , Εξοδος="26")
μήνυμα 'Ο Dekaxadikos 1A antistoixei ston dekadiko 26.'
- **M3 =>**
(Είσοδος = \$)
μήνυμα 'To psifio @ den einai egyro'
(Είσοδος = 1FFFFF , Εξοδος=2097151)
μήνυμα 'Ο Dekaxadikos 1FFFFF antistoixei ston dekadiko 2097151.'
- **M4 =>**
(Είσοδος = G)
μήνυμα 'To psifio @ den einai egyro'
- **M5 =>**
(Είσοδος = <)
μήνυμα 'To psifio @ den einai egyro'
- **M6 =>**
(Είσοδος = 123 , Εξοδος=291)
μήνυμα 'Ο Dekaxadikos 123 antistoixei ston dekadiko 291.'
- **M7 =>**
μήνυμα "dwste enan dekaexadiko arithmo (mexri 30 psifia) "
(Είσοδος = " ")
μήνυμα "Den dwsate kapoia psifia"
(Είσοδος = "1fA ")
μήνυμα 'To psifio f den einai egyro'
(Είσοδος = 1fA , Εξοδος=506)
μήνυμα 'Ο Dekaxadikos 1fA antistoixei ston dekadiko 506.'
- **M8 =>**
(Είσοδος = 1?3F)
μήνυμα 'To psifio ? den einai egyro'
- **M9 =>**
(Είσοδος = 1fA367 , Εξοδος=2073447)
μήνυμα 'Ο Dekaxadikos 1fA367 antistoixei ston dekadiko 2073447.
- **M10 =>**
(Είσοδος = 13D , Εξοδος=317)
μήνυμα 'Ο Dekaxadikos 13D antistoixei ston dekadiko 317.

Παρακάτω παρουσιάζονται μερικά στιγμιότυπα από λανθασμένα αποτελέσματα που εντοπιστήκαν με της παραπάνω περίπτωσης ελέγχου.

(Είσοδος = 0 , Αναμενόμενη Έξοδος=0)

```
Dwste enan dekaexadikos arithmo (mexri 30 psifia):0
0 Dekaxadikos '0' antistoixei ston dekadiko '-48'.
```

(Είσοδος = 1A , Αναμενόμενη Έξοδος=26)

```
Dwste enan dekaexadikos arithmo (mexri 30 psifia):1A
0 Dekaxadikos '1A' antistoixei ston dekadiko '112'.
```

(Είσοδος = 1FFFFFF , Αναμενόμενη Έξοδος=2097151)

```
Dwste enan dekaexadikos arithmo (mexri 30 psifia):1FFFFFF
0 Dekaxadikos '1FFFFFF' antistoixei ston dekadiko '16777152'.
```

(Είσοδος = 123 , Αναμενόμενη Έξοδος=291)

```
Dwste enan dekaexadikos arithmo (mexri 30 psifia):123
0 Dekaxadikos '123' antistoixei ston dekadiko '512'.
```

(Είσοδος = 1fa , Αναμενόμενη Έξοδος= 'Το psifio f den einai egyro')

```
Dwste enan dekaexadikos arithmo (mexri 30 psifia):1fa
To 1 psifio den einai egkyro!
```

(Είσοδος = G , Αναμενόμενη Έξοδος= 'Το psifio G den einai egyro')

```
Dwste enan dekaexadikos arithmo (mexri 30 psifia):G
To 0 psifio den einai egkyro!
```

(Είσοδος = -1 , Αναμενόμενη Έξοδος= άκυρη τιμή εισόδου)

```
Dwste enan dekaexadikos arithmo (mexri 30 psifia):-1
To 0 psifio den einai egkyro!
```

(Είσοδος = 13D , Αναμενόμενη Έξοδος=317)

```
Dwste enan dekaexadikos arithmo (mexri 30 psifia):13D
0 Dekaxadikos '13D' antistoixei ston dekadiko '928'.
```

Για να αποφύγουμε αυτά τα λάθη ο κώδικας θα πρέπει να διορθωθεί, παρακάτω παρουσιάζονται μια πιο βελτιωμένη έκδοση του κώδικα καθώς και μερικά στιγμιότυπα από την εκτέλεση του.

```
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <math.h>

// Function to convert hexadecimal to decimal
double hexToDecimal(const char hex[]) {
    double decimal = 0;
    int length = 0;

    // Calculate the length of the hex string
    while (hex[length] != '\0') {
        length++;
    }

    // Process each hex digit and convert to decimal
    for (int i = 0; i < length; i++) {
        char digit = hex[i];

        if (islower(digit)) {
            // Lowercase characters are not allowed
            printf("Error: Lowercase hexadecimal characters are not allowed.\n");
            exit(EXIT_FAILURE);
        }

        digit = toupper(digit);

        if (isdigit(digit)) {
            decimal += (digit - '0') * pow(16, length - i - 1);
        } else if (isxdigit(digit)) {
            decimal += (digit - 'A' + 10) * pow(16, length - i - 1);
        } else {
            // Invalid hex digit
            printf("Error: Invalid hexadecimal digit '%c'\n", hex[i]);
            exit(EXIT_FAILURE);
        }
    }

    return decimal;
}

int main() {
    char hex[31];

    while (1) {
        // Prompt user for input
        printf("Enter a hexadecimal number (up to 30 digits, uppercase): ");
```

```

if (scanf("%30s", hex) != 1) {
    // Empty input or invalid input format
    printf("Error: Invalid input format\n");
    fflush(stdin); // Clear input buffer
    continue;
}

// Check if the input contains any lowercase characters
int containsLowercase = 0;
for (int i = 0; hex[i] != '\0'; i++) {
    if (islower(hex[i])) {
        containsLowercase = 1;
        break;
    }
}

if (containsLowercase) {
    printf("Error: Lowercase hexadecimal characters are not allowed.\n");
    continue;
}

// Convert hex to decimal
double decimal = hexToDecimal(hex);

// Check if the result is within the limits of a double
if (isfinite(decimal)) {
    // Display the result
    printf("Hexadecimal: %s\n", hex);
    printf("Decimal: %.15f\n", decimal);
    break;
} else {
    // Result exceeds the limits of a double
    printf("Error: Result exceeds the limits of a double\n");
    exit(EXIT_FAILURE);
}
}

return 0;
}

```

Παρακάτω παρατηρούμε ότι εκτυπώνονται μόνο τα 30 ψηφία του δεκαεξάδικου αριθμού

```

Enter a hexadecimal number (up to 30 digits, uppercase): 23456789ABCDEF123456789ABCDEF245
Hexadecimal: 23456789ABCDEF123456789ABCDEF2
Decimal: 183138079419255074510533838619279360.000000000000000

```

Εδώ βλέπουμε την περίπτωση γράμματος που δεν είναι κεφαλαίου

```

Enter a hexadecimal number (up to 30 digits, uppercase): 1fA
Error: Lowercase hexadecimal characters are not allowed.
Enter a hexadecimal number (up to 30 digits, uppercase): █

```

Διάγραμμα ροής του προγράμματος

