

ΠΥ-255, Άνοιξη 2023 Τμή. Επιστ. Υπολογιστών Πανεπιστήμιο Κρήτης Λιθόδοκων Α. Μπλάας	Όνομα:  ΑΜ: 	Τελική Εξέταση Δευτέρα 29 Ιανουαρίου 2023, 09:30 - 12:30 Διαγωνισμός με ΚΛΕΙΣΤΕΣ σημειώσεις, Απουσιάζετε σε όλες τις ερωτήσεις, ΤΑ ΘΕΜΑΤΑ ΕΙΝΑΙ ΣΥΝΤΟΜΑ
--	--	---

Ερώτηση Α [60]

1. [5] Ποια από τα παρακάτω είναι σωστά words (δεσμευμένες λέξεις) της C; (i) <code>#else</code> (ii) <code>FOR</code> (iii) <code>extern</code> (iv) <code>printf</code> (v) <code>main</code> (vi) <code>#include</code> (vii) <code>case</code> (viii) <code>true</code> (ix) <code>malloc</code> (x) <code>sizeof</code>					
2. [10] (a) [5] Τι θα επιστρέψει η κλήση της <code>strlen()</code> με παράμετρο τις παρακάτω μεταβλητές; (i) <code>char *p = "12"</code> ; (ii) <code>char s = '1'</code> ; (iii) <code>char *s = NULL</code> ; (iv) <code>char s[3] = {'1', '2', '\0'}</code> ; (v) <code>char s[10] = {'1', '2', '\0'}</code> ; (b) [5] Τι θα επιστρέψει το <code>sizeof</code> για τις ίδιες μεταβλητές;					
3. [20] Δίνονται οι δυοπλανές δηλώσεις μεταβλητών. (a) [5] Περιγράψτε τον τύπο της κάθε μεταβλητής (<code>c1, c2, s1, s2, s3</code>). (b) [9] Δώστε την μορφή που έχει στη μνήμη η κάθε μεταβλητή και τα περιεχόμενά της μετά την αρχικοποίηση. (c) [3] Είναι ο τύπος των μεταβλητών <code>s1, s3</code> ισοδύναμοι; Εξηγήστε σύντομα. (d) [3] Τι θα τυπώσουν οι παρακάτω εντολές; (Αν κάποια εντολή είναι λάθος, εξηγήστε γιατί)	<pre>char c1 = 'a'; char *c2 = "a"; char s1[128] = "this is a string"; char *s2 = "this is a string"; char s3[1][128];</pre>				
(i) <code>printf("%c\n", c1);</code> (ii) <code>printf("%s\n", c2);</code> (iii) <code>printf("%s\n", s1);</code> (iv) <code>printf("%s\n", s2);</code> (v) <code>strcpy(s3[0], s1, 128); printf("%s\n", s3[0]);</code>					
4. [10] Δίνεται το δυοπλανό πρόγραμμα που αποτελείται από δύο αρχεία <code>f.h</code> και <code>f.c</code> . (a) [5] Ποιο είναι το output του C pre-processor; (b) [3] Πόσες φορές θα γίνει <code>include</code> το <code>f.h</code> στο <code>f.c</code> και σε ποιο σημείο/σημεία (μπορείτε να χρησιμοποιήσετε αριθμούς γραμμής); Εξηγήστε σύντομα. (c) [2] Αν δώσουμε το output του cpp στον compiler, θα παράγει το εκτελέσιμο a.out; Εξηγήστε σύντομα.	<table border="1"> <thead> <tr> <th>f.h</th> <th>f.c</th> </tr> </thead> <tbody> <tr> <td> <pre>#ifndef _F_H_ #define _F_H_ extern int %; #define C 10 #endif</pre> </td> <td> <pre>#define INC #include "f.h" #include "f.h" int main(void) { return % + C + __LINE__; }</pre> </td> </tr> </tbody> </table>	f.h	f.c	<pre>#ifndef _F_H_ #define _F_H_ extern int %; #define C 10 #endif</pre>	<pre>#define INC #include "f.h" #include "f.h" int main(void) { return % + C + __LINE__; }</pre>
f.h	f.c				
<pre>#ifndef _F_H_ #define _F_H_ extern int %; #define C 10 #endif</pre>	<pre>#define INC #include "f.h" #include "f.h" int main(void) { return % + C + __LINE__; }</pre>				
5. [15] Ποια είναι τα λογικά λάθη στα δύο παρακάτω τμήματα προγραμμάτων;					
(a) [3] <code>int get_prev_elt(int a, int n) { int prev; assert(--n >= 0); prev = a[n]; return prev; }</code>	(b) [4] <code>char *concat(const char *s1, const char *s2) { char s[MAX_SZ]; assert((strlen(s1)+strlen(s2)) <= MAX_SZ); strcpy(s, s1); strcpy(s+strlen(s1), s2); return s; }</code>				
(c) [4] <code>char c[10], *s; const char cc[10] = "ABC"; s = c; c = s; s[2] = cc[2]; cc[3] = s[2];</code>	(d) [4] <code>struct S {struct S *next; void *v;}; void free_list(struct S *n) { while (n) { free(n); free(n->v); n = n->next; } }</code>				

Ερώτηση Β [60]

1. [5] Τι τύπο πρέπει να έχει η μεταβλητή "i" ώστε τα δύο κομμάτια (1), (2) να είναι ισοδύναμα? Γιατί; (1) <code>while (i > 0) {printf("Message\n"); i--;} (2) <code>while (i) {printf("Message\n"); i--;} </code></code>	
2. [10] Έστω τα κλάσματα $(-a)/b$ και $-(a/b)$, όπου τα a, b είναι unsigned integers. (a) [5] Έχουν τα δύο κλάσματα πάντα την ίδια τιμή; Εξηγήστε. (b) [5] Πότε έχουν τα δύο αυτά κλάσματα την ίδια τιμή;	
3. [25] Γράψτε μια πολυμορφική συνάρτηση <code>init2d()</code> η οποία παίρνει σαν παράμετρο ένα διδιάστατο array με στοιχεία οποιουδήποτε τύπου και αρχικοποιεί το κάθε στοιχείο χρησιμοποιώντας μια συνάρτηση <code>initelt()</code> . Η συνάρτηση <code>initelt()</code> δίνεται από το πρόγραμμα που καλεί την <code>init2d()</code> . Χρησιμοποιείστε στην <code>init2d()</code> οποιεσδήποτε άλλες παραμέτρους είναι απαραίτητες για την υλοποίησή σας. (a) [5] Δώστε το prototype της συνάρτησης <code>init2d()</code> και των παραμέτρων της. (b) [10] Δώστε μια υλοποίηση της συνάρτησης <code>init2d()</code> . (c) [4] Δώστε ένα παράδειγμα κλήσης της <code>init2d()</code> χρησιμοποιώντας ένα διδιάστατο array <code>A[5][10]</code> από integers, όπου ο κάθε integer αρχικοποιείται σε μια τυχαία τιμή.	

(d)	[6] Δώστε ένα παράδειγμα κλήσης της <code>Init2d()</code> χρησιμοποιώντας ένα δισδιάστατο array <code>B[15][9]</code> από structs του τύπου <code>struct S {char *s; int v;}</code> όπου το κάθε struct αρχικοποιείται σε (NULL, τυχαίος integer).
4.	[20] Ορίστε το interface ενός αφηρημένου τύπου δεδομένων (ADT), <code>bigint.h</code> , για πράξεις με αριθμούς οποιουδήποτε μεγέθους και αριθμού ψηφίων και σε οποιαδήποτε βάση. Π.χ. θα μπορεί να αναπαραστήσει έναν δυαδικό αριθμό ή έναν δεκαδικό αριθμό ή έναν αριθμό σε βάση την μέγιστη τιμή ενός integer και με το κάθε ψηφίο να είναι ένας integer. Το interface θα πρέπει να δίνει την δυνατότητα να δηλώνει κανείς μεταβλητές που περιέχουν μεγάλους αριθμούς, να τους αρχικοποιεί, και να κάνει πρόσθεση και πολλαπλασιασμό.
(a)	[4] Δώστε τον αδιαφανή τύπο της δήλωσης για μεταβλητές του ADT και δείξτε πως θα δήλωνε ένα πρόγραμμα δύο μεγάλους αριθμούς <code>X, Y</code> .
(b)	[6] Δώστε την υλοποίηση του αδιαφανή τύπου. Μπορείτε να επιλέξετε τον τρόπο που θα αποθηκεύετε τα ψηφία του κάθε μεγάλου αριθμού, π.χ. σαν ένα array τόσων ακεραίων, όσα τα ψηφία. Εξηγήστε σύντομα τις όποιες επιλογές σας για το πως αποθηκεύετε τον αριθμό και σε ποια βάση γίνεται η αναπαράστασή του.
(c)	[6] Δώστε τα prototypes των συναρτήσεων για την δημιουργία και καταστροφή μεγάλων αριθμών. Κατά την δημιουργία ενός μεγάλου αριθμού θα μπορεί να γίνει και αρχικοποίησή του σε μια τιμή οποιουδήποτε μεγέθους. Εξηγήστε σύντομα.
(d)	[4] Δώστε τα prototypes των συναρτήσεων για τις πράξεις πρόσθεση και πολλαπλασιασμό. Εξηγήστε σύντομα.

Ερώτηση Γ [60]

1.	[45] Δίνεται το διπλανό πρόγραμμα <code>test.c</code> , το οποίο κάνουμε compile στο object file <code>test.o</code> και στη συνέχεια παράγουμε το εκτελέσιμο <code>a.out</code> .	<pre>#include <stdlib.h> #define SZ 1000 float A[SZ]; static char s[128]; unsigned long L = 2000; static float Fx; static float g(float x, int y){ float w; w = x * y; A2: return w; } void f(float a, int b){ int i; for (i=0; i<SZ; i++){ A[i] = g(a,b); } return; } int main(void){ char *sp; s[0] = '\0'; Fx = 1.0; sp = malloc(128); A1: f(Fx, L); return 0; }</pre>
(a)	[10] Δώστε την εικόνα της μνήμης του προγράμματος στο σημείο <code>A1</code> της εκτέλεσής του. Σημειώστε σε ποιο μέρος της μνήμης βρίσκεται η κάθε μεταβλητή, συνάρτηση, και όνομα του προγράμματος, συμπεριλαμβανομένων και των labels <code>A1, A2</code> .	
(b)	[14] Σχεδιάστε την εικόνα της στοίβας την πρώτη φορά που η εκτέλεση φτάνει στο σημείο <code>A2</code> . Σημειώστε όλες τις μεταβλητές που τοποθετούνται στη στοίβα με τις τιμές τους και όλες τις άλλες τιμές που τοποθετούνται στη στοίβα σε συστήματα <code>x86-32</code> .	
(c)	[10] Δώστε τα σύμβολα που υπάρχουν στο αρχείο <code>test.o</code> για τους σκοπούς του linking. Για κάθε σύμβολο σημειώστε τις πληροφορίες που του αντιστοιχούν και που χρειάζονται για τους σκοπούς του linking.	
(d)	[5] Δώστε την κλήση της συνάρτησης <code>f()</code> στην <code>main</code> όπως την παράγει ο compiler σε εντολές συστημάτων <code>x86-32</code> , συμπεριλαμβάνοντας τις «δουλειές» που χρειάζεται να γίνουν στην <code>main</code> πριν και μετά την κλήση της <code>f()</code> .	
(e)	[6] Δώστε με ακρίβεια δύο τρόπους για να μάθουμε την διεύθυνση των μεταβλητών <code>A</code> και <code>w</code> . Θα αλλάξει η διεύθυνση των μεταβλητών αυτών από εκτέλεση σε εκτέλεση του προγράμματος; Εξηγήστε σύντομα.	
2.	[15] Δίνεται το διπλανό πρόγραμμα που αποτελείται από ένα αρχείο.	<pre>#include <stdio.h> #include <setjmp.h> jmp_buf e; int in=0, out=0; void incr(int x){ in++; if (x<5){ x++; incr(x); } longjmp(e, x); out++; return; }</pre>
(a)	[10] Τι θα τυπώσει η <code>printf()</code> που βρίσκεται στην <code>main()</code> ;	<pre>int main(void){ int x=0, r; r = setjmp(e); if (r == 0) incr(x); else x=r; printf("x=%d, in=%d, out=\n%d\n", x, in, out); return 0; }</pre>
(b)	[3] Πόσα stack frames θα δημιουργηθούν στην στοίβα του προγράμματος για την αναδρομική κλήση της <code>incr()</code> ;	
(c)	[2] Τι μετράει η μεταβλητή <code>out</code> ; Τι τιμή έχει το <code>out</code> στην <code>printf()</code> και γιατί;	