Submit response?

Your username (**pp0783@srmist.edu.in**) and responses will be recorded when you submit this form.

SWITCH ACCOUNT SUBMIT

Part A-CT-2: 21CSS101J Programming for Problem Solving -CS Cognitive Computing/CS Cyber Security and Digital Forensics/CS Data Science/Artificial Intelligence

Date:11-12-2021
Time :8Am To 10.40Am
Part-A: 20 MCQ : 20 X 1 =20 marks
Part-B: 2 Big questions (either or type): 2 X 15 = 30 marks
Dr.R.Radhika

pp0783@srmist.edu.in Switch account

Draft restored

Your email will be recorded when you submit this form

* Required

Reg No: *

RA2112704010015

Name: *

PONNURI ANIRUDDHA

Department *

○ CS Cognitive Computing

○ CS Cyber Security and Digital Forensics

● CS Data Science

○ Artificial Intelligence

An unrestricted use of ‘goto’ statement is harmful because *

○ it results in increasing the executing time of the program

○ it increases the memory of the program

● it decreases the readability and testing of program

○ Increase in execution time

Consider the C program fragment below which is meant to divide x by y using repeated subtractions. The variables x, y, q and r are all unsigned int. while (r >= y) {r = r – y;q = q + 1;} Which of the following conditions on the variables x, y, q and r before the execution of the fragment will ensure that the loop terminates in a state satisfying the condition x == (y*q + r)? *

○ (q == r) && (r == 0)

○ (x > 0) && (r == x) &&(y > 0)

● (q == 0) && (r == x) && (y > 0)

○ (q == 0) && (y > 0)

What is the output of the program? void main ( ) {int a[ ] = {5,10,15,20,25} ,*i, *j;i = &a[0];j = &a[4]; printf(“%d, %d”, j–i,*j–*i);} *

○ 20, 4

○ 15, 20

○ 5, 25

● 4, 20

What would be the output of the following program? sum = 0; for (i = –10; i < 0; i++) sum = sum + abs(i); printf ("%d", sum); *

○ 100

● 55

○ -505

○ -55

The following function computes the maximum value contained in an integer array P [ ] of size n (n > = 1). int max (int *p, int n) { int a = 0, b = n – 1; while (_____) { if (p [a] < = p [b]) {a = a+1;} else { b = b – 1;} } return p[a]; } The missing loop condition is *

○ a ! = n

● b ! = 0

○ b > (a +1)

○ b ! = a

int ∗ S[a] is 1D array of integers, which of the following refers to the third element in the array? *

● ∗(S + 2)

○ ∗(S + 3)

○ S + 2

○ S + 3

If an array is declared as char a[10][12]; what is referred to by a[5]? *

○ Pointer to 3rd Row

● Pointer to 4th Row

○ Pointer to 5th Row

○ Pointer to 6th Row

How many times will the following code be executed?{x = 10;while (x = 1)x ++;} *

○ Never

○ Once

○ 15 times

● Infinite number of times

If the condition is missing in a FOR loop of a C program then *

● It is assumed to be present and taken to be false

○ It is assumed to be present and taken to be true **

○ It results in syntax error

○ Execution will be terminated abruptly

What is the output? main () {int a = 0; int b = 20; char x = 1; char y = 10;i f(a, b, x, y); printf(“hello”);} *

○ logical error

○ Garbage value

● hello

○ 20

Consider the C functions foo and bar given below: int foo (int val) {int x = 0; while (val > 0) {x =x + foo (val––);} return val; } int bar (int val) {int x = 0; while (val > 0) {x =x +bar (val – 1);} return val;} Invocations of foo (3) and bar (3) will result in: *

○ Return of 6 and 6 respectively.

○ Infinite loop and abnormal termination respectively

○ Abnormal termination and infinite loop respectively.

● Both terminating abnormally

Consider the following C function in which size is the number of elements in the array E:int MyX (int *E, unsigned int size) {int Y = 0;int Z;int i, j, k; for (i = 0; i < size; i++)Y = Y + E[i]; for (i = 0; i < size; i++) for (j = 1; j < size; j++){Z = 0; for (k = i; k < = j; k++)Z = Z + E[k]; if (Z > Y)Y = Z;}return Y;} The value returned by the function My X is the *

○ maximum possible sum of elements in any sub -array of array E.

○ maximum element in any sub-array of array E.

● sum of the maximum elements in all possible sub-arrays of array E.

○ the sum of all the elements in the array E.

Consider the following recursive C function void get (int n) { if (n < 1) return; get (n – 1); get (n – 3); printf(“%d”, n); } If get (6) function is being called in main ( ) then how many times will the get ( ) function be invoked before returning to the main ( )? *

○ 15

● 25

○ 45

○ 35

Output of the following C program is intF(int x, int *py, int **pz) {int y, z;** pz+= 1; z = *pz;*py+= 2; y = *py;x+ = 3; return x+y+z;} void main( ) {int c, *b, **a ; c = 4; b = &c; a = &b; printf(“%d”, F(c, b, a));} *

○ 30

○ 22

○ 22

○ 20

● Error

The following code is run from the command line as myprog 1 2 3. What would be the output? main(int argc, char *argv[ ]) { int i, j = 0; for (i = 1; i < argc; i++) j = j + atoi (argv [i]); printf (“%d”, j);} *

● 123

○ 6 **

○ Error

○ “123”

Consider the following C program: int f(int n) {static int r; if (n<=0) return 1; if (n> 3){r=n; return (f(n–2)+2));} return f(n–1) + r;} What is the value of f(5)? *

○ 15

○ 17

● 18

○ 19

What will be the output of following code? # include <stdio.h> aaa() {printf(“hi”);} bbb() {printf(“hello”);} ccc(){printf(“bye”);} main () {int *ptr[3]( ); ptr[0] = aaa; ptr[1] = bbb; ptr[2] = ccc; ptr[2]();} *

○ hi

○ hello

● bye

○ Garbage value

○ Garbage value

What will be the output of the following C program? void count (int n) {static int d = 1; printf(“%d ”,n); printf(“%d ”,d); d ++; if (n > 1) count (n -1); printf(“%d ”, d);} void main ( ) {count (3);} *

○ 3 1 2 2 1 3 4 4 4

● 3 1 2 1 1 1 2 2 2

○ 3 1 2 2 1 3 4

○ 3 1 2 1 1 1 2

Consider the following C code: # include <stdio.h> int *assignval (int *x, int val) {*x = val;return x;} void main ( ) {int *x = malloc (sizeof (int)); if (NULL == x) return; x = assignval (x, 0); if (x) {x = (int *) malloc(sizeof (int)); if (NULL == x) return; x = assignval (x, 10);} printf(“%d\n”, *x); free (x);} The code suffers from which one of the following problems: *

○ compiler error as the return of malloc is not typecast appropriately

● compiler error because the comparison should be made as x == NULL and not as shown

○ compiles successfully but execution may result in dangling pointer

○ compiles successfully but execution may result in memory leak

The value printed by the following program is ___. void f (int* p, int m) {m = m +5; *p = *p + m;return;} void main () {int i = 5, j = 10; f(&i, j); print f (“%d”, i +j);} *

○ 15

○ 20

● 30

○ 10

Submit Clear form