

Trình độ: ĐẠI HỌC; Ngày thi: 01/08/2022

Môn: CẤU TRÚC DỮ LIỆU

ĐÁP ÁN ĐỀ THI CHÍNH THỨC

(Đáp án - thang điểm gồm 05 trang)

Câu	Phần	Nội dung	Thang điểm
1		<pre>#include <stdio.h> #include <stdlib.h> struct node { int data; struct node *next; }; struct node *head = NULL; struct node *creatNode(int x) { struct node *p = (struct node *)malloc(sizeof(struct node)); p->data = x; p->next = NULL; return p; }</pre>	0.5 điểm
		<pre>void insert(int n) { struct node *p; int data; for (int i = 1; i <= n; i++) { printf("Nhap so node thu %d: ", i); scanf("%d", &data); p = creatNode(data); p->next = head; head = p; } }</pre>	0.5 điểm
		<pre>void printList() { struct node *p = head; if (p == NULL) printf("\n-----DANH SACH RONG-----\n\n"); else { while (p != NULL) { printf("%d ->", p->data); } } }</pre>	0.5 điểm

	<pre> p = p->next; } } </pre>	
	<pre> int countNodes() { int count = 0; struct node *p; p = head; while (p != NULL) { count++; p = p->next; } return count; } int main() { int n, Total; printf("Nhap tong so node: "); scanf("%d", &n); insert(n); printf("***** IN DANH SACH LIEN KET*****\n"); printList(); Total = countNodes(); printf("\n Tong so node trong DSLK = %d\n", Total); return 0; } </pre>	0.5 điểm
2	<pre> #include <stdio.h> #define max 100 int a[max], top = -1; int isStackEmpty() { if (top == -1) return 1; return 0; } int isStackFull() { if (top == max - 1) return 1; return 0; } </pre>	0.5 điểm
	<pre> void push(int x) { if (isStackFull()) printf("-----KHONG Push %d vi Stack DAY---\n", x); else { ++top; } } </pre>	0.5 điểm

		<pre> a[top] = x; } } int pop() { return a[top]; } </pre>	
		<pre> void converDecimaToBinary(int n, int radix) { int sodu = 0; while (n > 0) { sodu = n % radix; push(sodu); n /= radix; } printf("So nhi phan: "); for (int i = top; top > -1; top--) printf("%d", pop()); } void main() { int n, sodu = 0, item; printf("Nhap so thap phan: "); scanf("%d", &n); converDecimaToBinary(n, 2); } </pre>	1 điểm
3	a	<pre> #include <stdbool.h> #include <stdio.h> #include <stdlib.h> struct node { int key; struct node *left; struct node *right; }; struct node *newNode(int x) { struct node *p = (struct node *)malloc(sizeof(struct node)); p->key = x; p->left = NULL; p->right = NULL; return p; } struct node *insert(struct node *node, int x) { if (node == NULL) return newNode(x); </pre>	1 điểm

		<pre> if (x < node->key) node->left = insert(node->left, x); else if (x > node->key) node->right = insert(node->right, x); return node; } </pre>	
	b	<pre> void Nlr(struct node *root) { if (root != NULL) { printf("%d ->", root->key); Nlr(root->left); Nlr(root->right); } } </pre>	1 điểm
	b	<pre> void Lnr(struct node *root) { if (root != NULL) { Lnr(root->left); printf("%d ->", root->key); Lnr(root->right); } } </pre>	1 điểm
	b	<pre> void Lrn(struct node *root) { if (root != NULL) { Lrn(root->left); Lrn(root->right); printf("%d ->", root->key); } } </pre>	1 điểm
	c	<pre> int DEMLA(struct node *root) { if (root == NULL) return 0; else if (root->left == NULL && root->right == NULL) return 1; else return DEMLA(root->left) + DEMLA(root->right); } </pre>	1 điểm
	d	<pre> int MaxNode(struct node *node) { if (node->right == NULL) return node->key; else </pre>	1 điểm

	<pre> return MaxNode(node->right); } </pre>	
	<pre> int main() { struct node *root = NULL; int dem = 0; root = insert(root, 50); insert(root, 30); insert(root, 20); insert(root, 40); insert(root, 70); insert(root, 60); insert(root, 80); printf("Duyet cay theo thu tu LNR\n"); Lnr(root); printf("\nDuyet cay theo thu tu NLR\n"); Nlr(root); printf("\nDuyet cay theo thu tu LRN\n"); Lrn(root); printf("\nSo nut la: %d\n", DEMLA(root)); printf("Nut lon nhat trong cay: %d", MaxNode(root)); return 0; } </pre>	
TỔNG ĐIỂM		10