演習１

ソースプログラム

#include <stdio.h> /\* 標準入出力のヘッダファイルの読込み \*/

#include <stdlib.h> /\* 標準ライブラリのヘッダファイルの読込み \*/

int gcd(int a0, int a1); /\* 関数の宣言 \*/

main()

/\* メインプログラム：最大公約数計算のためのデータの入出力 \*/

{

int a0, a1, temp; /\* 変数の宣言 \*/

printf("Type in the first integer.\n"); /\* a0の入力を指示 \*/

scanf("%d", &a0); /\* キーボードからa0の入力 \*/

printf("a0 = %d\n", a0); /\* a0の出力 \*/

printf("Type in the second integer.\n"); /\* a1の入力を指示 \*/

scanf("%d", &a1); /\* キーボードからa1の入力 \*/

printf("a1 = %d\n", a1); /\* a1の出力 \*/

if(a0<=0 || a1<=0) /\* 正性のチェック \*/

{printf("Illegal input number\n"); exit(1);}

if(a0<a1) {temp=a0; a0=a1; a1=temp;} /\* a0>=a1に正規化 \*/

printf("GCD = %d\n", gcd(a0,a1)); /\* 最大公約数の計算と出力 \*/

return(0);

}

int gcd(int a0, int a1) /\* ユークリッドの互除法 \*/

/\* ユークリッドの互除法: a0とa1の最大公約数の計算 \*/

{

int a, b, temp; /\* 変数の宣言 \*/

a=a0; b=a1; /\* 初期設定 \*/

while(b!=0) {temp=a%b; a=b; b=temp;} /\* 互除法の反復 \*/

return(a); /\* 計算結果を返す \*/

}

実行結果

C:\Users\yutow\OneDrive\デスクトップ\データ構造演習12,19>cl gcd.cユークリッド.c

Microsoft(R) C/C++ Optimizing Compiler Version 19.38.33133 for x64

Copyright (C) Microsoft Corporation. All rights reserved.

gcd.cユークリッド.c

Microsoft (R) Incremental Linker Version 14.38.33133.0

Copyright (C) Microsoft Corporation. All rights reserved.

/out:gcd.cユークリッド.exe

gcd.cユークリッド.obj

C:\Users\yutow\OneDrive\デスクトップ\データ構造演習12,19>gcd.cユークリッド

Type in the first integer.

377

a0 = 377

Type in the second integer.

203

a1 = 203

GCD = 29

演習２の後順

ソースプログラム/\* 後順による木のなぞりのプログラム例 \*/

#include <stdio.h> /\* 標準入出力のヘッダファイルの読込み \*/

#include <stdlib.h> /\* 標準ライブラリのヘッダファイル \*/

#define N 100 /\* 配列Sの最大サイズ（節点数） \*/

struct cell /\* 構造体cellの定義 \*/

{

int node;

struct cell \*next;

};

void postorder(int k, struct cell \*\*S); /\* 関数の宣言 \*/

main()

/\* Traverse a tree S in postrder. \*/

{

struct cell \*S[N], \*p, \*q;

int n, j, parent, child, root;

FILE \*file;

file=fopen("treedata.txt", "r");

fscanf(file, "%d", &n);

if(n>N)

{

printf("Illegal array size n = %d for N = %d\n", n, N);

exit(1);

}

printf("n = %d\n", n);

for(j=0; j<n; j++) S[j]=NULL;

fscanf(file, "%d", &parent);

while(parent >= 0)

{

fscanf(file, "%d", &child);

if(child >= 0)

{

p=(struct cell \*)malloc(sizeof(struct cell));

S[parent]=q=p;

}

while(child >= 0)

{

q->node=child;

q->next=NULL;

fscanf(file, "%d", &child);

if(child >= 0)

{

p=(struct cell \*)malloc(sizeof(struct cell));

q->next=p;

q=p;

}

}

fscanf(file, "%d", &parent);

}

for(j=0; j<n; j++)

{

printf("S[%d] = %p, ", j, S[j]);

q=S[j];

while(q != NULL)

{

printf("node = %d, next = %p, ", q->node, q->next);

q=q->next;

}

printf("\n");

}

root=0;

printf("postorder =");

postorder(root, S);

printf("\n");

return(0);

}

void postorder(int k, struct cell \*\*S)

/\* S[k]を根とする部分木の後順によるなぞり \*/

{

struct cell \*q;

q=S[k]; /\* kから前順のなぞり \*/

while(q != NULL)

{

postorder(q->node, S); /\* 再帰呼び出し \*/

q=q->next;

}

printf(" %d", k); /\* 節点kの出力 \*/

return;

}

実行結果

C:\Users\yutow\OneDrive\デスクトップ\データ構造演習12,19>cl 後順による木のなぞりのプログラム例.c

Microsoft(R) C/C++ Optimizing Compiler Version 19.38.33133 for x64

Copyright (C) Microsoft Corporation. All rights reserved.

後順による木のなぞりのプログラム例.c

Microsoft (R) Incremental Linker Version 14.38.33133.0

Copyright (C) Microsoft Corporation. All rights reserved.

/out:後順による木のなぞりのプログラム例.exe

後順による木のなぞりのプログラム例.obj

C:\Users\yutow\OneDrive\デスクトップ\データ構造演習12,19>後順による木のなぞりのプログラム例

n = 10

S[0] = 00000208FD1E0720, node = 1, next = 00000208FD1E0700, node = 2, next = 0000000000000000,

S[1] = 00000208FD1E0740, node = 3, next = 00000208FD1E0640, node = 4, next = 00000208FD1E0660, node = 5, next = 0000000000000000,

S[2] = 00000208FD1E08A0, node = 6, next = 00000208FD1E07C0, node = 7, next = 0000000000000000,

S[3] = 0000000000000000,

S[4] = 0000000000000000,

S[5] = 0000000000000000,

S[6] = 00000208FD1E0840, node = 8, next = 00000208FD1E08E0, node = 9, next = 0000000000000000,

S[7] = 0000000000000000,

S[8] = 0000000000000000,

S[9] = 0000000000000000,

postorder = 3 4 5 1 8 9 6 7 2 0

演習２の中順

ソースプログラム

/\* 中順による木のなぞりのプログラム例 \*/

#include <stdio.h> /\* 標準入出力のヘッダファイルの読込み \*/

#include <stdlib.h> /\* 標準ライブラリのヘッダファイル \*/

#define N 100 /\* 配列Sの最大サイズ（節点数） \*/

struct cell /\* 構造体cellの定義 \*/

{

int node;

struct cell \*next;

};

void inorder(int k, struct cell \*\*S); /\* 関数の宣言 \*/

main()

/\* Traverse a tree S in inorder. \*/

{

struct cell \*S[N], \*p, \*q;

int n, j, parent, child, root;

FILE \*file;

file=fopen("treedata.txt", "r");

fscanf(file, "%d", &n);

if(n>N)

{

printf("Illegal array size n = %d for N = %d\n", n, N);

exit(1);

}

printf("n = %d\n", n);

for(j=0; j<n; j++) S[j]=NULL;

fscanf(file, "%d", &parent);

while(parent >= 0)

{

fscanf(file, "%d", &child);

if(child >= 0)

{

p=(struct cell \*)malloc(sizeof(struct cell));

S[parent]=q=p;

}

while(child >= 0)

{

q->node=child;

q->next=NULL;

fscanf(file, "%d", &child);

if(child >= 0)

{

p=(struct cell \*)malloc(sizeof(struct cell));

q->next=p;

q=p;

}

}

fscanf(file, "%d", &parent);

}

for(j=0; j<n; j++)

{

printf("S[%d] = %p, ", j, S[j]);

q=S[j];

while(q != NULL)

{

printf("node = %d, next = %p, ", q->node, q->next);

q=q->next;

}

printf("\n");

}

root=0;

printf("inorder =");

inorder(root, S);

printf("\n");

return(0);

}

void inorder(int k, struct cell \*\*S)

/\* S[k]を根とする部分木の中順によるなぞり \*/

{

struct cell \*q;

q=S[k]; /\* kから前順のなぞり \*/

while(S !=NULL){

S=S++;

}

while(q !=NULL){

printf(" %d", k); /\* 節点kの出力 \*/

q->next;

}

inorder(q->node, S); /\* 再帰呼び出し \*/

return;

}

実行結果

C:\Users\yutow\OneDrive\デスクトップ\データ構造演習12,19>cl 中順による木のなぞりのプログラム例.c

Microsoft(R) C/C++ Optimizing Compiler Version 19.38.33133 for x64

Copyright (C) Microsoft Corporation. All rights reserved.

中順による木のなぞりのプログラム例.c

C:\Users\yutow\OneDrive\デスクトップ\データ構造演習12,19\中順による木のなぞりのプログラム例.c(90) : warning C4717: 'inorder': すべて のコントロールのパス、関数を回帰するとランタイム スタック オーバーフローが発生します。

Microsoft (R) Incremental Linker Version 14.38.33133.0

Copyright (C) Microsoft Corporation. All rights reserved.

/out:中順による木のなぞりのプログラム例.exe

中順による木のなぞりのプログラム例.obj

C:\Users\yutow\OneDrive\デスクトップ\データ構造演習12,19>中順による木のなぞりのプログラム例

n = 10

S[0] = 00000147022A06A0, node = 1, next = 00000147022A0740, node = 2, next = 0000000000000000,

S[1] = 00000147022A0920, node = 3, next = 00000147022A07A0, node = 4, next = 00000147022A0720, node = 5, next = 0000000000000000,

S[2] = 00000147022A06E0, node = 6, next = 00000147022A0760, node = 7, next = 0000000000000000,

S[3] = 0000000000000000,

S[4] = 0000000000000000,

S[5] = 0000000000000000,

S[6] = 00000147022A0640, node = 8, next = 00000147022A0980, node = 9, next = 0000000000000000,

S[7] = 0000000000000000,

S[8] = 0000000000000000,

S[9] = 0000000000000000,

inorder =