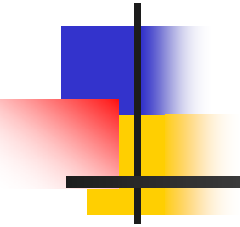


# Ampliación de la Programación



## 2 Procesadores Programación Dinámica



# 2 Procesadores: Prog. Dinámica

## Resolución

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

```
#define MAX 20
```

```
#define INFINITO 1000
```

```
#define TRUE 1
```

```
#define FALSE 0
```

```
typedef struct {
```

```
    int tiempo;
```

```
    int trabajo [MAX]; // vector[i] es del procesador que hace el trabajo i
```

```
} TipoSolucion;
```

```
void Procesadores (int Etapa, int tiempo0, int tiempo1,  
                  TipoSolucion *solOptima, int N, int Coste[2][MAX]);
```

```
void Salida (TipoSolucion sol, int N, int Coste[2][MAX]);
```

```
int maximo (int a, int b) {
```

```
    if (a < b) return b ;else return a;
```

```
}
```



# 2 Procesadores: Prog. Dinámica

## Resolución

```
int main () {  
    int Coste [2][MAX]; // matriz de costes  
    int N, i, j;  
    TipoSolucion solOptima;  
  
    // entrada de datos  
    printf("Introduzca el número de trabajos "); scanf("%d",&N);  
    printf("\nDuración de los trabajos en el procesador 0 (separados por un espacio)\n");  
    for (i=0; i<N; i++) scanf(" %d", &Coste[0][i]);  
    printf("\nDuración de los trabajos en el procesador 1 (separados por un espacio)\n");  
    for (i=0; i<N; i++)  
        scanf(" %d", &Coste[1][i]);  
    for (i= 0; i<2; i++) {  
        for (j=0; j<N; j++)  
            printf(" %4d",Coste[i][j]);  
        printf("\n");  
    }  
  
    Procesadores(0,0,0,&solOptima,N,Coste);  
    Salida(solOptima,N,Coste);  
    return 0;  
}
```



# 2 Procesadores: Prog. Dinámica

## Resolución

```
void Procesadores (int k, int tiempo0, int tiempo1,
                  TipoSolucion *solOptima, int N, int Coste[2][MAX]) {
    TipoSolucion sol0, sol1;

    if (k == N)
        solOptima->tiempo = maximo (tiempo0, tiempo1);
    else {
        // lo hace el procesador 0
        Procesadores (k+1, tiempo0+Coste[0][k], tiempo1, &sol0, N, Coste);
        sol0.trabajo[k] = 0;

        // lo hace el procesador 1
        Procesadores (k+1, tiempo0, tiempo1+Coste[1][k], &sol1, N, Coste);
        sol1.trabajo[k] = 1;

        // nos quedamos con la mejor de las dos
        if (sol0.tiempo < sol1.tiempo) *solOptima = sol0;
        else *solOptima = sol1;
    }
}
```



# 2 Procesadores: Prog. Dinámica

---

## Resolución

```
void Salida (TipoSolucion sol, int N, int Coste[2][MAX]) {  
    int i;  
  
    printf("\nTrabajos realizados por el procesador 0: ");  
    for (i=0; i<N; i++)  
        if (sol.trabajo[i] == 0) printf("%d (+ %d ) ", i, Coste[0][i]);  
  
    printf("\nTrabajos realizados por el procesador 1: ");  
    for (i=0; i<N; i++)  
        if (sol.trabajo[i] == 1) printf("%d (+ %d ) ", i, Coste[1][i]);  
    printf("\n\nEl coste de la solucion es: %d", sol.tiempo);  
  
}
```