

Oefenzitting 4 - Functies II (Rijen)

1.10 Functies in Drama, rijen als parameter

Oefening 1

Teken de activatie-records voor alle functies en stel de toekenningstabellen op. Vertaal daarna de code en teken de stapel zoals hij eruit ziet bij de **tweede** recursieve oproep van `quickSort` die zelf de functie `partition` opgeroepen heeft (bij het pijltje).

```
int partition(int r[], register int left, register int right) {
    int pivot;
    register int tmp;
    // <-- Stapel
    pivot = r[left];
    while (left <= right) {
        while (r[left] < pivot)
            left++;
        while (r[right] > pivot)
            right--;
        if (left <= right) {
            tmp = r[left];
            r[left] = r[right];
            r[right] = tmp;
            left++;
            right--;
        }
    }
    return left;
}

void quickSort(int r[], int start, int end) {
    int idx;
    idx = partition(r, start, end);
    if (start < idx - 1)
        quickSort(r, start, idx - 1);
    if (end > idx)
        quickSort(r, idx, end);
}
```

```

int rij[] = {3, 1, 5, 8, 7, 2, 4, 9, 6, 0};
main() {
    register int * ptr;
    quicksort(rij, 0, 9);
    for (ptr = &rij[0]; ptr < &rij[9]; ptr)
        printint(*ptr++);
}

```

Oefening 2

Vertaal het volgende C-programma dat de Fibonacci-getallen berekent. Teken de stapel zoals ze er uit ziet net voor de terugkeer uit de functie `fibon`.

```

void fibon(int n, int r[]) {
    register int i;
    r[0] = 1;
    r[1] = 1;
    for (i=2; i<n; i++)
        r[i] = r[i-1] + r[i-2];
}

int f[100];
main () {
    register int i;
    int m;
    m = getint();
    if (m < 100) {
        fibon(m, f); /* bereken fibonacci getallen */
        for(i=0; i<m; i++)
            printint(f[i]); /* druk fibonacci getallen */
    }
}

```

Oefening 3

Vertaal het volgende C-programma. Teken de stapel zoals ze eruit ziet net na de toekenning van `*r` wanneer `idx = 3` en `max = 10`.

```

void recFibon(register int max, register int idx, register int * r) {
    register int i;
    *r = r[-1] + r[-2]; // M0: r[-1] == *(r - 1)
    if (++idx <= max)
        recFibon(max, idx, r+1);
}

```

```

void fibon(register int max, register int r[]) {
    r[0] = r[1] = 1;
    if (max > 1)
        recFibon(max, 2, &r[2]);
}

int f[100];
main () {
    register int i;
    int m;
    m = getint();
    if (m < 100) {
        fibon(m, f); /* bereken fibonacci getallen */
        for(i=0; i<m; i++)
            printint(f[i]); /* druk fibonacci getallen */
    }
}

```

Oefening 4

Vertaal het volgende C programma. Beschrijf in woorden wat het doet.

```

void lees(int n, int r[]) {
    register int i;
    for (i=0; i < n; i++)
        r[i] = getint();
}

void schrijfOmgekeerd(register int n, register int *r) {
    register int i;
    r += n - 1;
    for (i=0; i < n; i++)
        printint(*r--);
}

main () {
    int rij[5];
    register int aantal;
    aantal = getint();
    if (aantal <= 5) {
        lees(aantal, rij);
        schrijfOmgekeerd(aantal, rij);
    }
}

```