

1

Peactical NO 8

Aim: - write a c program for selection sort Generate equivalent 3 add-code.

Tools :

SE.HO	Tools	specification	aby
Ŧ	C compilet	Tubboc, ale, any -	-
2	Computer System	windows, 49BRAM-	1

Theory :-

Selection sort: - It's a simple and easy to understand sorting algorithm that works by repeatedly selecting the smallest element from unsorted pattern of the list.

It's repeated for the remaining unsorted pattern portion of the list untile the entire list is sorted. Three address code. It's a type of intermediate code which is easy to generales and can be easily converted to machines code it makes use of atmost three address code and one operator to represent an expression

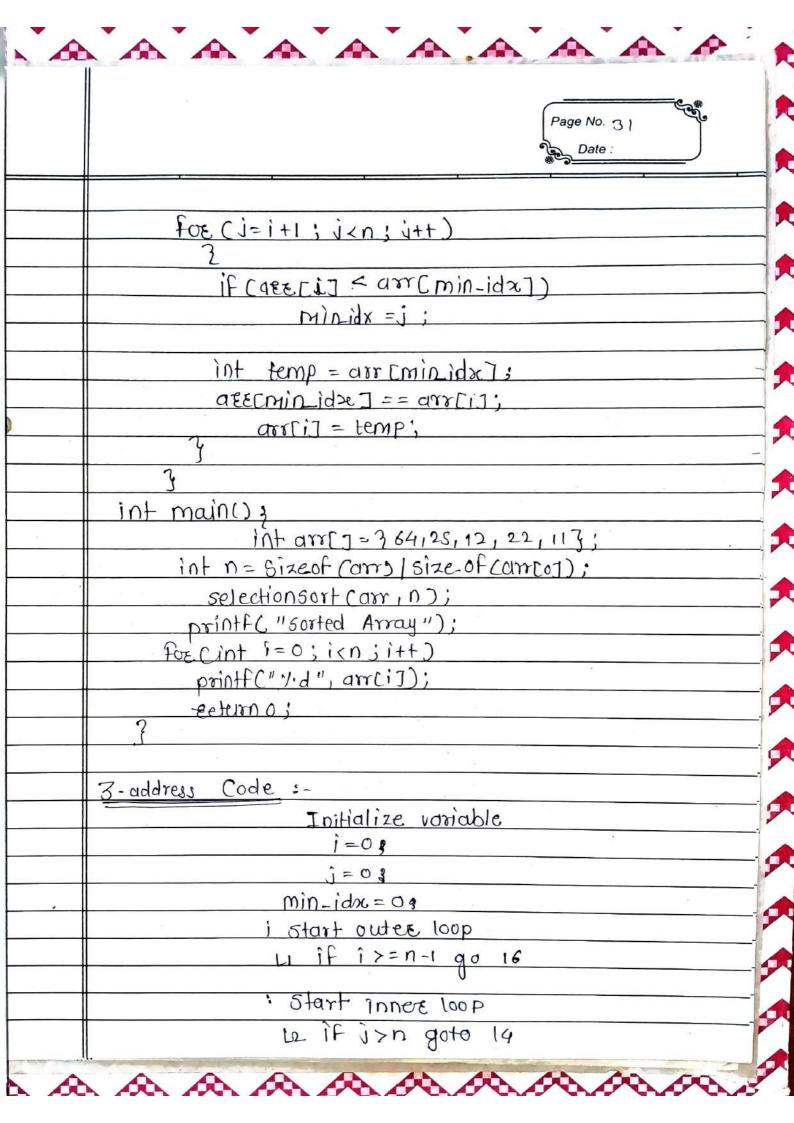
Program: #include < stdio. h>

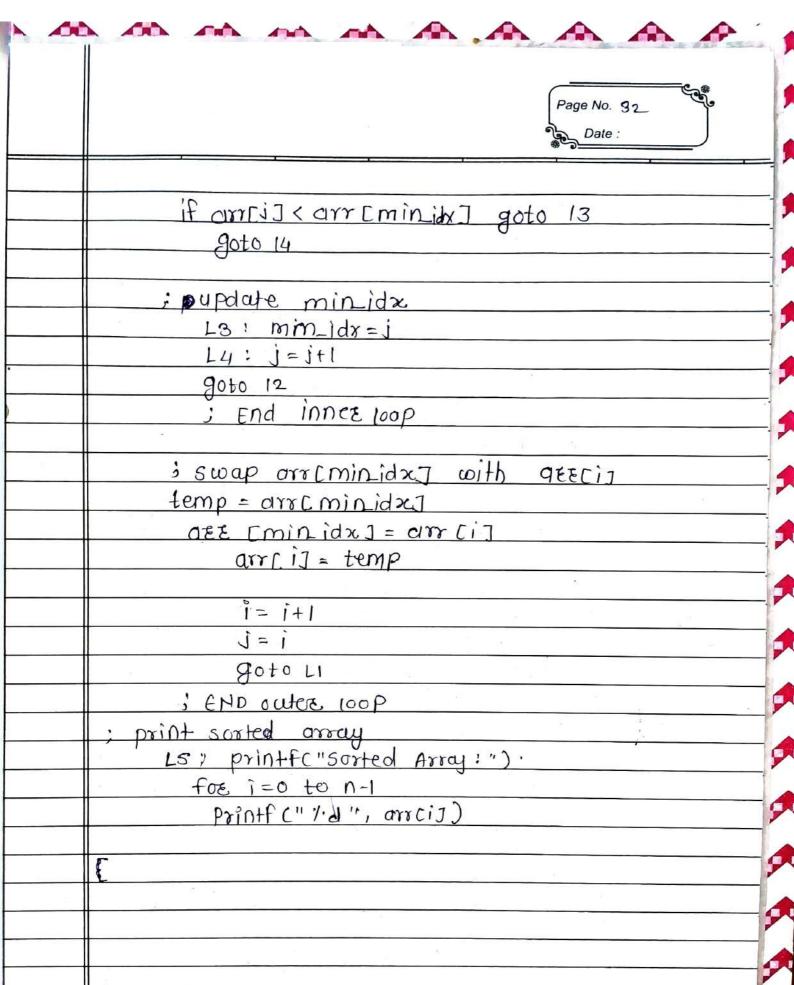
void · Gelectionsort Cint arres, int H)

int i, j, min_Pdx;

fez (i=0; i<n-1; i++)

m in - idx = i;

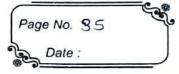




A A A A A A A A A A A A A A

	Page No. 33 Date:
ERRORS and Remedial a	ction:
ERRORS: Hot handling	dep
Remedial Action :- we keep touch of all occurrer electent.	are modify algorithm laces of the minimum
Conclusion: C program for equivalent 3 add code how successfully.	re selection sort and its
Cource outcomes attained Program outcomes	8 mapping with
. cource outcome	Peogram outcome
19ACC to syntaxe divided teanslation for generating intermediate rode 3-add code	Poz: Problem Analysis Poz: Problem Analysis Poz: Problem Analysis Solution.

	Peachical NO 09	Page No.	
1	Izîte a Program î	n c peam o prog ress rode gene	rated.
TOULS :-	Toole	s pecification	Wy
1	compilere	€ computer (any) turbo c++ c++	-
2	computer system	4GB RAM Windows, ITB HDD	1
generations one operations one operations	intermediate, re e, and can be e it make use of reator to represent computed at each ry variable general	easily converted of at most saddring an expression is	to Machiness and and the sorted in
interme phases of Code of	mization:- 3 Address diate representation of the comilation allows the compiles Optimization that of generated code	process the to analyze the can improve the	optimiza 3-address and



PROGRAM: Optimized Three Address 1) i=0 2) j=0 3) min_indoc 4) if i>n-1 goto (23) 5] if i>=n goto (43) c) t1 = arr Cj] g) t2 = arr Cmin_ind J k) if t1 <t2, +3z="" 1)="" 12]="" c10)="" c13)="" goto="" i="+3</th" j+1;="" min_indocj;=""><th>code:</th></t2,>	code:
Optimized Three Address 1) i=0 2) j=0 3) min_indoc 4) if i>n-1 goto (23) 5] if i>=n goto (43) c) t1 = arr (j) 2) t2 = arr (min_ind) (a) if t1 <t2, (10)="" (13)="" +3z="" 1)="" 2)="" goto="" j+1;<="" min_indxj;="" td=""><td>code:</td></t2,>	code:
1) i=0 2) j=0 3) min_indoc 4) if i>n-1 goto (23) 5] if i>=n goto (43) 6) t1 = arr (i) 7) t2 = arr (min_ind] 8) if t1 <t2, (10)="" (13)="" +3z="" 1)="" 9)="" goto="" j+1;<="" min_indx;;="" td=""><td></td></t2,>	
2) j=0 3) min_indoc 4) if i>n-1 goto (23) 5] if i>=n goto (43) 6) t1 = arr (j) 2) t2 = arr (min_ind) 6) if t1 <t2, (10)="" (13)="" +32="" 1)="" goto="" j+1;<="" min_indxj;="" td=""><td></td></t2,>	
3) min_indoc 4) if i>n-1 goto (23) 5] if i>=n goto (43) c> t1 = arr (i) 2) t2 = arr (min_ind) (i) if t1 <t2, (10)="" (13)="" +32="" 1)="" 2)="" goto="" j+1;<="" min_indx;;="" td=""><td></td></t2,>	
4) if i>n-1 goto (23) 5) if i>=n goto (43) 6) t1 = arr (i) 7) t2 = arr (min-ind) 8) if t1 <t2, (10)="" (13)="" +32="" 1)="" 9)="" goto="" j+1;<="" min_indx;;="" td=""><td></td></t2,>	
4) if i>n-1 goto (23) 5) if i>=n goto (43) 6) t1 = arr (i) 7) t2 = arr (min-ind) 8) if t1 <t2, (10)="" (13)="" +32="" 1)="" 9)="" goto="" j+1;<="" min_indx;;="" td=""><td></td></t2,>	
() t1 = arr (J] 2) t2 = arr (min-ind] () if t1 <t2, (10)<br="" goto="">() goto (13) () min_indx;; () +32 j+1;</t2,>	
() t1 = arr (J] 2) t2 = arr (min-ind] () if t1 <t2, (10)<br="" goto="">1) goto (13) 1) min_indx;; () +3z j+1;</t2,>	
(j) if \$1 <t2, (10)="" (13)="" +3z="" 1)="" goto="" j+1;<="" min_indx;;="" td="" y)=""><td></td></t2,>	
1) goto (13) 1) min_indxj; 1) +3z j+1;	
1) min_indxj; 1) +3z j+1;	
V) +32 j+1;	
$ 12 i = \pm 3$	
13) 9060	
14) sevap to with arocij	
15] t4 = arr[i]	2.
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	
(18) $t4 = t5$	e e e e e e e e e e e e e e e e e e e
13) to = 161	
$ 20\rangle$ $\hat{j} = \pm 6$	
j = i	
22) got 0 ()	
23) Printf ("sorted Array").	2
24) for 61=0 to n-1 goto (257	
28] printf (113/0d", tu)	
28) goto (24)	
U U	

AAAAAAAA

	Page No. 36 Date:			
EROP ERROR and Remedi	al Adion_			
Error: - Not handling a	Implicate elements correctly			
Remedial: We can mo	odify algorithm to keep			
and sutup them all to co	of the minimum elemen			
program to implement c program for				
Selection sout, then generating three address for it, and after this optimizing the three				
code and generation of objection	ext code implemented			
successfully.	eer cood miplanamen			
U	a a			
Course outcome attained o	ind mapping with pois:			
Course outcome	Program outcomes			
co-4: Build a code generation	Poi: Engineering knowledge			
using different intermediate	Poi: Engineering knowledge. Po2: problem Analyze			
and code and optimize	PO3: Design development of			
the target code.	Solution			
	POJ: Moderen tool usage			
	POG: individual and			
	team work.			

Page No. 37 PEachical NO 10 Aim: write a c program for selection sort generate and generate objects rode for three address code. Tools :specification aby SR. NO TOOLS 1. compuler tubboc, C++ 14CC 2. computer system windows, 4GBRAM ITB HDD Theory :-Three address code is on intermediate code it is used by the optimizing compilers In 3-address code the given expression is heaken down into several separate instructions these instruction can easily translate into assembly language such three address code intendion has at most three operands. It is a combination of assignment and a binary operator PROGRACH: - object code Generated from optimized 3-address code: start: mor i, o 3) mor j, 0 4) 100p outer s] compln, i, imp 100p 6) comple n, j, imp (11) 7) mor are ELIJ, +1 MANA MA MA MA A

P

P

1

1

